Ab initio solution of the many-electron Schrödinger equation with deep neural networks

David Pfau,[*,†] James S. Spencer,[*] and Alexander G. D. G. Matthews
DeepMind, 6 Pancras Square, London N1C 4AG, United Kingdom

W. M. C. Foulkes
Department of Physics, Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom

(Received 18 March 2020; accepted 6 August 2020; published 16 September 2020)

Given access to accurate solutions of the many-electron Schrödinger equation, nearly all chemistry could be derived from first principles. Exact wave functions of interesting chemical systems are out of reach because they are NP-hard to compute in general, but approximations can be found using polynomially scaling algorithms. The key challenge for many of these algorithms is the choice of wave function approximation, or Ansatz, which must trade off between efficiency and accuracy. Neural networks have shown impressive power as accurate practical function approximators and promise as a compact wave-function Ansatz for spin systems, but problems in electronic structure require wave functions that obey Fermi-Dirac statistics. Here we introduce a novel deep learning architecture, the Fermionic neural network, as a powerful wave-function Ansatz for many-electron systems. The Fermionic neural network is able to achieve accuracy beyond other variational quantum Monte Carlo Ansatz on a variety of atoms and small molecules. Using no data other than atomic positions and charges, we predict the dissociation curves of the nitrogen molecule and hydrogen chain, two challenging strongly correlated systems, to significantly higher accuracy than the coupled cluster method, widely considered the most accurate scalable method for quantum chemistry at equilibrium geometry. This demonstrates that deep neural networks can improve the accuracy of variational quantum Monte Carlo to the point where it outperforms other *ab initio* quantum chemistry methods, opening the possibility of accurate direct optimization of wave functions for previously intractable many-electron systems.

DOI: 10.1103/PhysRevResearch.2.033429

I. INTRODUCTION

The success of deep learning in artificial intelligence [1,2] has led to an outpouring of research into the use of neural networks for quantum physics and chemistry. Many of these methods train a deep neural network to predict properties of novel systems by use of supervised learning on a dataset compiled from existing computational methods—typically density functional theory (DFT) [3,4], exact solutions on a lattice [5], or coupled cluster with single, double and perturbative triple excitations [CCSD(T)] [6,7]. Yet all of these methods have drawbacks. Even CCSD(T), which is generally much more accurate than DFT, has difficulties with bond breaking and transition states [8]. While methods exist that are even more accurate, most suffer from impractical scaling (in the worst case exponential) [9] or require complicated system-dependent tuning, making them difficult to apply "out-of-the-box" to new systems. Here we focus instead on *ab initio* methods that use deep neural networks as approximate solutions to the many-electron Schrödinger equation *without the need for external data*. We are able to achieve very high accuracy on a number of small but challenging systems, all with the same neural network architecture, suggesting that our method could be a promising "out-of-the-box" solution for larger systems for which existing approaches are insufficient.

The ground-state wave function $\psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ and energy E of a chemical system with n electrons may be found by solving the time-independent Schrödinger equation,

$$\hat{H}\psi(\mathbf{x}_1, \ldots, \mathbf{x}_n) = E\psi(\mathbf{x}_1, \ldots, \mathbf{x}_n), \tag{1}$$

$$\hat{H} = -\frac{1}{2}\sum_i \nabla_i^2 + \sum_{i>j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{iI}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \sum_{I>J}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|},$$

where $\mathbf{x}_i = \{\mathbf{r}_i, \sigma_i\}$ are the coordinates of electron i, with $\mathbf{r}_i \in \mathbb{R}^3$ the position and $\sigma_i \in \{\uparrow, \downarrow\}$ the spin, ∇_i^2 is the Laplacian with respect to $\mathbf{r}_i$, and $\mathbf{R}_I$ and Z_I are the position and atomic number of nucleus I. We work in the Born-Oppenheimer approximation [10], where the nuclear positions are fixed input parameters, and Hartree atomic units are used throughout. The Schrödinger differential operator is spin independent

[*]These authors contributed equally to this work.
[†]pfau@google.com

but the electron spin matters because the wave function must obey Fermi-Dirac statistics—it must be antisymmetric under the simultaneous exchange of the position and spin coordinates of any two electrons: $\psi(\ldots,\mathbf{x}_i,\ldots,\mathbf{x}_j,\ldots) = -\psi(\ldots,\mathbf{x}_j,\ldots,\mathbf{x}_i,\ldots)$.

Many approaches in quantum chemistry start from a finite set of one-electron orbitals $\phi_1,\ldots,\phi_N$ and approximate the many-electron wave function as a linear combination of antisymmetrized tensor products (Slater determinants) of those functions:

$$\sum_{\mathcal{P}} \operatorname{sign}(\mathcal{P}) \prod_i \phi_i^k(\mathbf{x}_{\mathcal{P}_i}) = \begin{vmatrix} \phi_1^k(\mathbf{x}_1) & \ldots & \phi_1^k(\mathbf{x}_n) \\ \vdots & & \vdots \\ \phi_n^k(\mathbf{x}_1) & \ldots & \phi_n^k(\mathbf{x}_n) \end{vmatrix} = \det\left[\phi_i^k(\mathbf{x}_j)\right] = \det[\mathbf{\Phi}^k], \quad (2)$$

$$\psi(\mathbf{x}_1,\ldots,\mathbf{x}_n) = \sum_k \omega_k \det[\mathbf{\Phi}^k], \quad (3)$$

where $\{\phi_1^k,\ldots,\phi_n^k\}$ is a subset of n of the N orbitals, the sum in Eq. (3) is taken over all permutations $\mathcal{P}$ of the electron indices, and the sum in Eq. (4) is over all subsets of n orbitals. The difficulty is that the number of possible Slater determinants rises exponentially with the system size, restricting this "full configuration-interaction" (FCI) approach to tiny molecules, even with recent advances [11].

To address problems of practical interest, a more compact representation of the wave function is needed. The choice of function class used to approximate the wave function is known as the wave-function Ansatz. For most applications of quantum Monte Carlo (QMC) methods to quantum chemistry, the default choice is the Slater-Jastrow Ansatz [12], which takes a truncated linear combination of Slater determinants and adds a multiplicative term—the Jastrow factor—to capture close-range correlations. The Jastrow factor is normally a product of functions of the distances between pairs and triplets of particles. Additionally, a backflow transformation [13] is sometimes applied before the orbitals are evaluated, shifting the position of every electron by an amount dependent on the positions of nearby electrons. There are many alternative Ansatz [14,15], but for continuous-space many-electron problems in three dimensions the Slater-Jastrow-backflow form remains the default.

Here, we greatly improve the accuracy of the Slater-Jastrow-backflow variational quantum Monte Carlo (VMC) method by using a neural network we dub the Fermionic Neural Network, or FermiNet, as a more flexible Ansatz. This avoids the use of a finite basis set, a significant source of error for other Ansatz, and models higher-order electron-electron interactions compactly. The use of neural networks as a compact wave-function Ansatz has been studied before for spin systems [16–20] and many-electron systems on a lattice [19,21] as well as small systems of bosons in continuous space [22]. Applications of neural network Ansatz to chemical systems have been limited to date, presumably due to the complexity of Fermi-Dirac statistics. Existing work has been restricted to very small numbers of electrons [23], or has been of very low accuracy [24]. Unlike these other approaches, we use the Slater determinant as the starting point for our Ansatz, and then extend it by generalizing the single-electron orbitals to include generic exchangeable nonlinear interactions of *all* electrons. In a conventional backflow transformation, the electron positions $\mathbf{r}_j$ at which the one-electron orbitals in the Slater determinants are evaluated are replaced by collective coordinates $\mathbf{r}_j + \sum_{i(\neq j)} \eta(r_{ij})(\mathbf{r}_i - \mathbf{r}_j)$, but the orbitals remain functions of a single three-dimensional variable. The FermiNet wave function goes much further, replacing the one-electron orbitals $\phi_i^k(\mathbf{x}_j)$ by functions of $3n$ independent variables. Every "orbital" in every determinant now depends both on $\mathbf{x}_j$ and (in a general symmetric way) on the position and spin coordinates of every other electron.

Our approach is similar in spirit to the neural network backflow transform [21] that has been applied to discrete systems. Certain simplifications in the discrete case allow the use of conventional neural networks, while the continuous case requires a novel architecture to handle antisymmetry constraints, boundary conditions and cusps. The closest prior work we are aware of in continuous space is the iterative backflow transform [25,26], which has been applied to superfluid ^{3}He. While that work uses intermediate layers of the same dimensionality as the input, the FermiNet can use intermediate layers of arbitrary dimensionality, increasing the representational capacity [27].

The FermiNet is not only an improvement over existing Ansatz for VMC, but is competitive with and in some cases superior to more sophisticated quantum chemistry algorithms. Projector methods such as diffusion quantum Monte Carlo (DMC) [12] and auxiliary field quantum Monte Carlo (AFQMC) [28] generate stochastic trajectories that sample the ground-state wave function without the need for an explicit representation, although accurate explicit trial wave functions are still required for good performance and numerical stability. We find the FermiNet is competitive with projector methods on all systems investigated, in contrast with the conventional wisdom that VMC is less accurate. Coupled cluster methods [8] use an Ansatz that multiplies a reference wave function by an exponential of a truncated sum of creation and annihilation operators. This proves remarkably accurate for equilibrium geometries, but conventional reference wave functions are insufficient for systems with many low-lying excited states. We evaluate the FermiNet on a variety of stretched systems and find that it outperforms coupled cluster in all cases.

II. FERMIONIC NEURAL NETWORKS

A. Fermionic neural network architecture

To construct an expressive neural network Ansatz, we note that nothing requires the orbitals in the matrix in Eq. (3) to be functions of the coordinates of a single electron. The only requirement for the determinant of a matrix-valued function of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ to be antisymmetric is that exchanging any two input variables, $\mathbf{x}_i$ and $\mathbf{x}_j$, exchanges two rows or columns of the output matrix, leaving the rest invariant. This observation allows us to replace the single-electron orbitals $\phi_i^k(\mathbf{x}_j)$ by multielectron functions $\phi_i^k(\mathbf{x}_j;\mathbf{x}_1,\ldots,\mathbf{x}_{j-1},\mathbf{x}_{j+1},\ldots,\mathbf{x}_n) = \phi_i^k(\mathbf{x}_j;\{\mathbf{x}_{/j}\})$, where $\{\mathbf{x}_{/j}\}$ denotes the set of all electron states except $\mathbf{x}_j$, so long as these functions are invariant to any change in the order of the arguments after $\mathbf{x}_j$. In theory, a

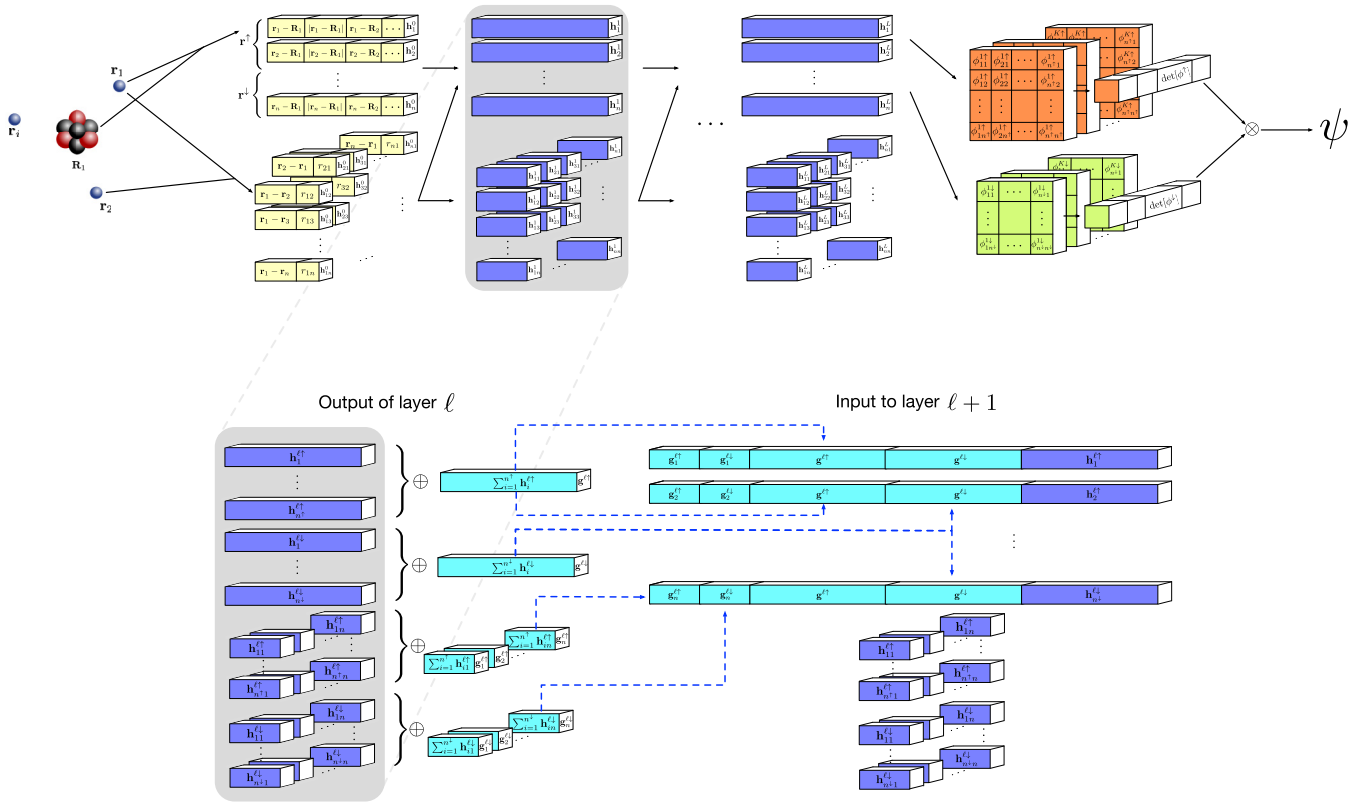

FIG. 1. The Fermionic neural network (FermiNet). Top: Global architecture. Features of one or two electron positions are inputs to different streams of the network. These features are transformed through several layers, a determinant is applied, and the wave function at that position is given as output. Bottom: Detail of a single layer. The network averages features of electrons with the same spin together, then concatenates these features to construct an equivariant function of electron position at each layer.

single determinant made up of these permutation-equivariant functions is sufficient to represent any antisymmetric function (see Appendix B); however, the practicality of approximating an antisymmetric function will depend on the choice of permutation-equivariant function class; we hence use a small linear combination of n_k determinants in this work. The construction of a set of these permutation-equivariant functions with a neural network is the main innovation of the FermiNet. We emphasize that determinants constructed from permutation-equivariant functions are substantially more expressive than conventional Slater determinants. Figure 1 contains a schematic of the network and Algorithm I pseudocode for evaluating the network.

The Fermionic neural network takes features of single electrons and pairs of electrons as input. As input to the single-electron stream of the network, we include both the difference in position between each electron and nucleus $\mathbf{r}_i - \mathbf{R}_I$ and the distance $|\mathbf{r}_i - \mathbf{R}_I|$. The input to the two-electron stream is similarly the differences $\mathbf{r}_i - \mathbf{r}_j$ and distances $|\mathbf{r}_i - \mathbf{r}_j|$. Adding the absolute distances between particles directly as input removes the need to include a separate Jastrow factor after a determinant. As the distance is a nonsmooth function at zero, the neural network is capable of expressing the nonsmooth behavior of the wave function when two particles coincide—the wave-function cusps. Accurately modeling the cusps is critical for correctly estimating the energy and other properties of the system. The quality of the wave-function cusps for the helium atom are investigated in Appendix F. We denote the concatenation of all features for one electron $\mathbf{h}_i^0$, or $\mathbf{h}_i^{0\alpha}$ if we explicitly index its spin $\alpha \in \{\uparrow, \downarrow\}$; the features of two electrons are denoted $\mathbf{h}_{ij}^0$ or $\mathbf{h}_{ij}^{0\alpha\beta}$. If the system has $n^\uparrow$ spin-up electrons and $n^\downarrow$ spin down electrons, then without loss of generality we can reorder the electrons so that $\sigma_j = \uparrow$ for $j \in 1, \ldots, n^\uparrow$ and $\sigma_j = \downarrow$ for $j \in n^\uparrow + 1, \ldots, n$.

To satisfy the overall antisymmetry constraint for a fermionic wave function, intermediate layers of the Fermionic Neural Network must mix information together in a permutation-equivariant way. Permutation-equivariant neural network layers like self-attention have gained success in recent years in natural language processing [29] and protein folding [30], but we pursue a simpler yet effective approach. Permutation-equivariant layers have also been widely adopted in the computational chemistry and machine learning community for modeling energies and force fields from atomic configurations [3,31,32]. The Fermionic Neural Network shares some architectural details with these models, such as the use of pairwise distances as inputs and parallel streams of feature vectors, one per particle, through the network, but is tailored specifically for mapping electronic configurations to wave-function values with fixed atomic positions, rather than mapping atomic positions to total energies and other properties.

In our intermediate layers, we take the mean of activations from different streams of the network, concatenate these mean

Algorithm 1: FermiNet evaluation.

Require: walker configuration $\{\mathbf{r}_1^\uparrow, \cdots, \mathbf{r}_{n^\uparrow}^\uparrow, \mathbf{r}_1^\downarrow, \cdots, \mathbf{r}_{n^\downarrow}^\downarrow\}$
Require: nuclear positions $\{\mathbf{R}_I\}$
for each electron i, α **do**
  $\mathbf{h}_i^{\ell\alpha} \leftarrow$ concatenate$(\mathbf{r}_i^\alpha - \mathbf{R}_I, |\mathbf{r}_i^\alpha - \mathbf{R}_I| \; \forall \, I)$
  $\mathbf{h}_{ij}^{\ell\alpha\beta} \leftarrow$ concatenate$(\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta, |\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta| \; \forall \, j, \beta)$
end for
for each layer $\ell \in \{0, L-1\}$ **do**
  $\mathbf{g}^{\ell\uparrow} \leftarrow \frac{1}{n^\uparrow} \sum_i^{n^\uparrow} \mathbf{h}_i^{\ell\uparrow}$
  $\mathbf{g}^{\ell\downarrow} \leftarrow \frac{1}{n^\downarrow} \sum_i^{n^\downarrow} \mathbf{h}_i^{\ell\downarrow}$
  for each electron i, α **do**
    $\mathbf{g}_i^{\ell\alpha\uparrow} \leftarrow \frac{1}{n^\uparrow} \sum_j^{n^\uparrow} \mathbf{h}_{ij}^{\ell\alpha\uparrow}$
    $\mathbf{g}_i^{\ell\alpha\downarrow} \leftarrow \frac{1}{n^\downarrow} \sum_j^{n^\downarrow} \mathbf{h}_{ij}^{\ell\alpha\downarrow}$
    $\mathbf{f}_i^{\ell\alpha} \leftarrow$ concatenate$(\mathbf{h}_i^{\ell\alpha}, \mathbf{g}^{\ell\uparrow}, \mathbf{g}^{\ell\downarrow}, \mathbf{g}_i^{\ell\alpha\uparrow}, \mathbf{g}_i^{\ell\alpha\downarrow})$
    $\mathbf{h}_i^{\ell+1\alpha} \leftarrow \tanh\left(\text{matmul}(\mathbf{V}^l, \mathbf{f}_i^{\ell\alpha}) + \mathbf{b}^l\right) + \mathbf{h}_i^{\ell\alpha}$
    $\mathbf{h}_{ij}^{\ell+1\alpha\beta} \leftarrow \tanh\left(\text{matmul}(\mathbf{W}^l, \mathbf{h}_{ij}^{\ell\alpha\beta}) + \mathbf{c}^l\right) + \mathbf{h}_{ij}^{\ell\alpha\beta}$
  end for
end for
for each determinant k **do**
  for each orbital i **do**
    for each electron j, α **do**
      $e \leftarrow$ envelope$(\mathbf{r}_j^\alpha, \{r_i^\alpha - \mathbf{R}_I\})$
      $\phi_i(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar\alpha}\}) = (\text{dot}(\mathbf{w}_i^{k\alpha}, \mathbf{h}_j^{L\alpha}) + g_i^{k\alpha})e$
    end for
  end for
  $D^{k\uparrow} \leftarrow \det\left[\phi_i^{k\uparrow}(\mathbf{r}_j^\uparrow; \{\mathbf{r}_{/j}^\uparrow\}; \{\mathbf{r}^\downarrow\})\right]$
  $D^{k\downarrow} \leftarrow \det\left[\phi_i^{k\downarrow}(\mathbf{r}_j^\downarrow; \{\mathbf{r}_{/j}^\downarrow\}; \{\mathbf{r}^\uparrow\})\right]$
end for
$\psi \leftarrow \sum_k \omega_k D^{k\uparrow} D^{k\downarrow}$

activations together and append them to the single-electron streams of the network. For a single layer this is a purely linear operation, but when combined with a nonlinear activation function after each layer it becomes a flexible architecture for building permutation-equivariant functions [33]. Information from both the other one-electron streams and the two-electron streams are fed into the one-electron streams. However, to reduce the computational overhead, no information is transferred between two-electron streams—these are multilayer perceptrons running in parallel. If the outputs of the one-electron stream at layer ℓ with spin α are denoted $\mathbf{h}_i^{\ell\alpha}$ and outputs of the two-electron stream are $\mathbf{h}_{ij}^{\ell\alpha\beta}$, then the input to the one-electron stream for electron i with spin α at layer $\ell+1$ is

$$\left(\mathbf{h}_i^{\ell\alpha}, \frac{1}{n^\uparrow}\sum_{j=1}^{n^\uparrow}\mathbf{h}_j^{\ell\uparrow}, \frac{1}{n^\downarrow}\sum_{j=1}^{n^\downarrow}\mathbf{h}_j^{\ell\downarrow}, \frac{1}{n^\uparrow}\sum_{j=1}^{n^\uparrow}\mathbf{h}_{ij}^{\ell\alpha\uparrow}, \frac{1}{n^\downarrow}\sum_{j=1}^{n^\downarrow}\mathbf{h}_{ij}^{\ell\alpha\downarrow}\right) = \left(\mathbf{h}_i^{\ell\alpha}, \mathbf{g}^{\ell\uparrow}, \mathbf{g}^{\ell\downarrow}, \mathbf{g}_i^{\ell\alpha\uparrow}, \mathbf{g}_i^{\ell\alpha\downarrow}\right) = \mathbf{f}_i^{\ell\alpha}, \tag{4}$$

which is the concatenation of the mean activation for spin up and down parts of the one and two electron streams, respectively. This concatenated vector is then input into a linear layer followed by a tanh nonlinearity. A residual connection is also added between layers of the same shape, for both one and two electron streams:

$$\begin{aligned}\mathbf{h}_i^{\ell+1\alpha} &= \tanh\left(\mathbf{V}^\ell \mathbf{f}_i^{\ell\alpha} + \mathbf{b}^\ell\right) + \mathbf{h}_i^{\ell\alpha},\\ \mathbf{h}_{ij}^{\ell+1\alpha\beta} &= \tanh\left(\mathbf{W}^\ell \mathbf{h}_{ij}^{\ell\alpha\beta} + \mathbf{c}^\ell\right) + \mathbf{h}_{ij}^{\ell\alpha\beta}.\end{aligned} \tag{5}$$

After the last intermediate layer of the network, a final spin-dependent linear transformation is applied to the activations, and the output is multiplied by a weighted sum of exponentially decaying envelopes, which enforces the boundary condition that the wave function goes to zero far away from the nuclei:

$$\begin{aligned}\phi_i^{k\alpha}\left(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar\alpha}\}\right) = \left(\mathbf{w}_i^{k\alpha} \cdot \mathbf{h}_j^{L\alpha} + g_i^{k\alpha}\right)\\ \times \sum_m \pi_{im}^{k\alpha} \exp\left(-\left|\mathbf{\Sigma}_{im}^{k\alpha}\left(\mathbf{r}_j^\alpha - \mathbf{R}_m\right)\right|\right),\end{aligned} \tag{6}$$

where $\bar\alpha$ is $\downarrow$ if α is $\uparrow$ or vice versa, $\mathbf{h}_j^{L\alpha}$ is an output from the Lth (final) layer of the intermediate single-electron features network for electrons of spin α, and $\mathbf{w}_i^{k\alpha}$ ($g_i^{k\alpha}$) are the weights (biases) of the final linear transformation for determinant k. The learned parameters $\pi_{im}^{k\alpha}$ and $\mathbf{\Sigma}_{im}^{k\alpha} \in \mathbb{R}^{3\times3}$ control the anisotropic decay to zero far from each nucleus. The functions $\{\phi_i^{k\alpha}(\mathbf{r}_j^\alpha)\}$ are then used as the input to multiple determinants, and the full wave function is taken as a weighted sum of these determinants:

$$\begin{aligned}\psi(\mathbf{r}_1^\uparrow, \ldots, \mathbf{r}_{n^\downarrow}^\downarrow) = \sum_k \omega_k \left(\det\left[\phi_i^{k\uparrow}(\mathbf{r}_j^\uparrow; \{\mathbf{r}_{/j}^\uparrow\}; \{\mathbf{r}^\downarrow\})\right]\right.\\ \left.\times \det\left[\phi_i^{k\downarrow}(\mathbf{r}_j^\downarrow; \{\mathbf{r}_{/j}^\downarrow\}); \{\mathbf{r}^\uparrow\}; \right]\right).\end{aligned} \tag{7}$$

Equation (8) uses the fact that the full determinant $\det[\mathbf{\Phi}] = \det\left[\phi_i(\mathbf{x}_j; \{\mathbf{x}_{/j}\})\right]$ may be replaced by a product of spin-up and spin-down terms if we choose $\phi_i(\mathbf{x}_j; \{\mathbf{x}_{/j}\}) = 0$ if $i \in 1 \ldots n^\uparrow$ and $j \in n^\uparrow + 1, \ldots, n$ or $i \in n^\uparrow + 1, \ldots, n$ and $j \in 1, \ldots, n^\uparrow$. Then the matrix $\mathbf{\Phi}$ is block-diagonal and

$$\begin{aligned}\det[\mathbf{\Phi}] &= \det[\phi_i(\mathbf{x}_j; \{\mathbf{x}_{/j}\})]\\ &= \det[\phi_i^\uparrow(\mathbf{r}_j^\uparrow; \{\mathbf{r}_{/j}^\uparrow\}; \{\mathbf{r}^\downarrow\})] \det[\phi_i^\downarrow(\mathbf{r}_j^\downarrow; \{\mathbf{r}_{/j}^\downarrow\}; \{\mathbf{r}^\uparrow\})].\end{aligned} \tag{8}$$

The new wave function is only antisymmetric under exchange of electrons of the same spin, $\{\mathbf{r}^\uparrow\}$ or $\{\mathbf{r}^\downarrow\}$, but nevertheless yields correct expectation values of spin-independent observables and the fully antisymmetric wave function can be reconstructed if required. This factorization allows spin-dependence to be handled explictly rather than as input to the network.

The linear combination of determinants in Eq. (8) bears some resemblance to Ansatz used in truncated configuration interaction methods like CI singles and doubles (CISD), which are known to have issues with size-consistency, thus it is natural to wonder if the FermiNet also has these issues. The determinants in the FermiNet are very different from conventional Slater determinants, as they allow for essentially arbitrary correlations between electrons in each orbital $\phi_i^{k\alpha}$. We prove in Appendix B that a single determinant of this form is in theory general enough to represent *any* antisymmetric function, though in practice we require a small number of determinants to reach high accuracy. This may be due to the limitations of finite-size neural networks in representing

functions of the type described in Appendix B. In all our experiments on N_2 and the hydrogen chain (Secs. III D and III E, Table VI), the FermiNet was able to learn a size-consistent solution.

B. Wave-function optimization

As in the standard setting for wave-function optimization for variational Monte Carlo, we sought to minimize the energy expectation value of the wave-function Ansatz:

$$\mathcal{L}(\theta) = \frac{\langle\psi_\theta|\hat{H}|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} = \frac{\int d\mathbf{X}\,\psi_\theta^*(\mathbf{X})\hat{H}\psi_\theta(\mathbf{X})}{\int d\mathbf{X}\,\psi_\theta^*(\mathbf{X})\psi_\theta(\mathbf{X})},$$

where θ are the parameters of the Ansatz, $\hat{H}$ is the Hamiltonian of the system as given in Eq. (1), and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ denotes the full state of all electrons. As $\hat{H}$ is time-reversal invariant and Hermitian, its eigenfunctions and eigenvalues are real. If the minimization is taken over all real normalizable functions, then the minimum of the energy occurs when $\psi_\theta(\mathbf{X})$ is the ground-state eigenfunction of $\hat{H}$; for a more restricted Ansatz, the minimum lies above the ground-state eigenvalue. When samples from the probability distribution defined by the wave-function Ansatz $p(\mathbf{X}) \propto \psi_\theta^2(\mathbf{X})$ are available, unbiased estimates of the gradient of the energy with respect to θ can be computed as follows:

$$\begin{aligned} E_L(\mathbf{X}) &= \psi^{-1}(\mathbf{X})\hat{H}\psi(\mathbf{X}), \\ \nabla_\theta\mathcal{L} &= \mathbb{E}_{p(\mathbf{X})}[(E_L - \mathbb{E}_{p(\mathbf{X})}[E_L])\nabla_\theta \log|\psi|], \end{aligned} \tag{9}$$

where E_L is the *local energy* and we have dropped the dependence of ψ on θ for clarity. Recent developments [28,35–37], including investigating first-order stochastic opitimization methods from the machine learning community [38,39], have enabled optimization of conventional wave functions with large parameter sets. We use a second-order method which can exploit the structure of the neural network.

For all wave-function Ansätze used in this paper, the determinants were computed in the log domain, and the final network output gave the log of the absolute value of the wave function, along with its sign. The local energy was computed directly in the log domain using the formula:

$$\begin{aligned} E_L(\mathbf{X}) &= \psi^{-1}(\mathbf{X})\hat{H}\psi(\mathbf{X}) \\ &= -\frac{1}{2}\sum_i \left[\frac{\partial^2 \log|\psi|}{\partial r_i^2}\bigg|_{\mathbf{X}} + \left(\frac{\partial \log|\psi|}{\partial r_i}\bigg|_{\mathbf{X}}\right)^2\right] + V(\mathbf{X}), \end{aligned}$$

where $V(\mathbf{X})$ is the potential energy of the state $\mathbf{X}$ and the index i runs over all 3N dimensions of the electron position vector. To optimize the wave function, we used a modified version of Kronecker-factored approximate curvature (KFAC) [40], an approximation to natural gradient descent [41] appropriate for neural networks. Natural gradient descent updates for optimizing $\mathcal{L}$ with respect to parameters θ have the form $\delta\theta \propto \mathcal{F}^{-1}\nabla_\theta\mathcal{L}(\theta)$, where $\mathcal{F}$ is the Fisher Information Matrix (FIM):

$$\mathcal{F}_{ij} = \mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial \log p(\mathbf{X})}{\partial\theta_i}\frac{\partial \log p(\mathbf{X})}{\partial\theta_j}\right].$$

This is equivalent to stochastic reconfiguration [42] when the probability density is unnormalized (see Appendix C) and closely related to the linear method of Toulouse and Umrigar [43].

For large neural networks with thousands to millions of parameters, solving the linear system $\mathcal{F}\delta\theta = \nabla_\theta\mathcal{L}$ becomes impractical. KFAC ameliorates this with two approximations. First, any terms $\mathcal{F}_{ij}$ are assumed to be zero when θ_i and θ_j are in different layers of the network. This makes the FIM block-diagonal and significantly more efficient to invert. The second approximation is based on the structure of the gradient for a linear layer in a neural network. If W_ℓ is the weight matrix for layer ℓ of a network, then the block of the FIM for that weight is, in vectorized form,

$$\begin{aligned} &\mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial \log p(\mathbf{X})}{\partial \mathrm{vec}(\mathbf{W}_\ell)}\frac{\partial \log p(\mathbf{X})}{\partial \mathrm{vec}(\mathbf{W}_\ell)}^T\right] \\ &\quad = \mathbb{E}_{p(\mathbf{X})}[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^T], \end{aligned} \tag{10}$$

where $\mathbf{a}_\ell$ are the forward activations and $\mathbf{e}_\ell$ are the backward sensitivities for that layer. KFAC approximates the inverse of this block as the Kronecker product of the inverse second moments:

$$\begin{aligned} \mathbb{E}_{p(\mathbf{X})}[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^T]^{-1} &\approx \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell\mathbf{a}_\ell^T]^{-1} \\ &\otimes \mathbb{E}_{p(\mathbf{X})}[\mathbf{e}_\ell\mathbf{e}_\ell^T]^{-1}. \end{aligned} \tag{11}$$

Further details can be found in Martens and Grosse (2015) [40].

The original KFAC derivation assumed the density to be estimated was normalized, but we wish to extend it to stochastic reconfiguration for unnormalized wave functions. In Appendix C, we show that if we only have access to an unnormalized wave function, terms in the FIM can be expressed as

$$\mathcal{F}_{ij} \propto \mathbb{E}_{p(\mathbf{X})}[(\mathcal{O}_i - \mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_i])(\mathcal{O}_j - \mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_j])],$$

where $\mathcal{O}_i = \frac{\partial \log|\psi|}{\partial x_i}$. The terms in the FIM for the weights of a linear neural network layer would then be

$$\begin{aligned} \mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial \log p(\mathbf{X})}{\partial \mathrm{vec}(\mathbf{W}_\ell)}\frac{\partial \log p(\mathbf{X})}{\partial \mathrm{vec}(\mathbf{W}_\ell)}^T\right] &\propto \mathbb{E}_{p(\mathbf{X})}[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell - \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell \otimes \mathbf{e}_\ell])(\mathbf{a}_\ell \otimes \mathbf{e}_\ell - \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell \otimes \mathbf{e}_\ell])^T] \\ &= \mathbb{E}_{p(\mathbf{X})}[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^T] - \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell \otimes \mathbf{e}_\ell]\mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell \otimes \mathbf{e}_\ell]^T. \end{aligned}$$

We use a similar approximation for the inverse to that of conventional KFAC, replacing the uncentered second moments with mean-centered covariances:

$$\mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial \log p(\mathbf{X})}{\partial \mathrm{vec}(\mathbf{W}_\ell)}\frac{\partial \log p(\mathbf{X})}{\partial \mathrm{vec}(\mathbf{W}_\ell)}^T\right] \approx \mathbb{E}_{p(\mathbf{X})}\left[\hat{\mathbf{a}}_\ell\hat{\mathbf{a}}_\ell^T\right]^{-1} \otimes \mathbb{E}_{p(\mathbf{X})}\left[\hat{\mathbf{e}}_\ell\hat{\mathbf{e}}_\ell^T\right]^{-1}, \tag{12}$$

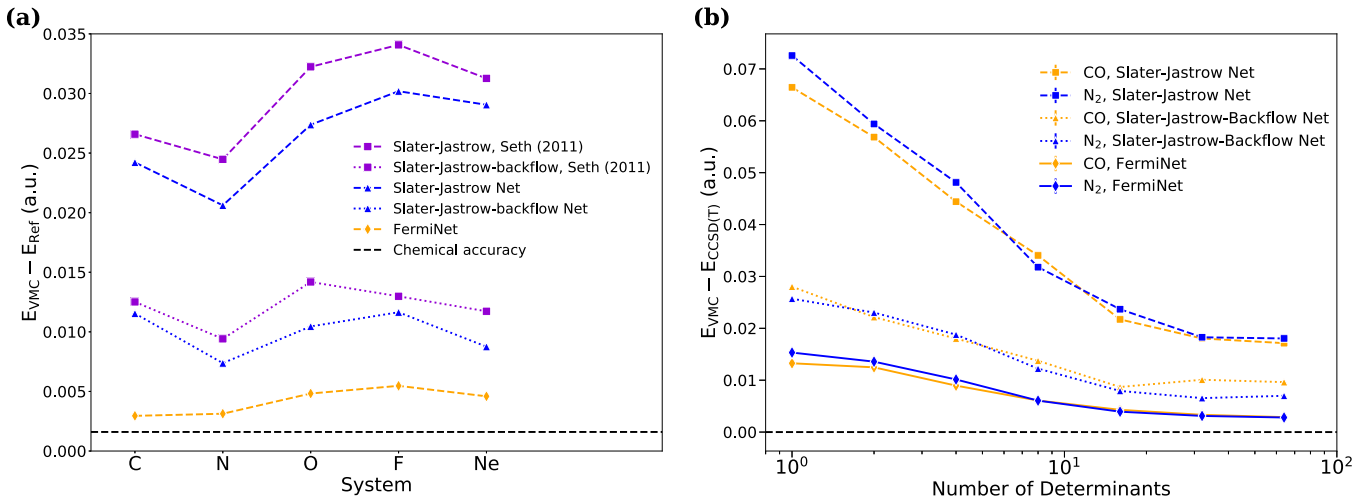

FIG. 2. Comparison of the FermiNet against the Slater-Jastrow Ansatz, with and without backflow. (a) First-row atoms with a single determinant. Baseline numbers are from Chakravorty *et al.* [34]. The Slater-Jastrow neural network yields slightly lower energies than VMC with a conventional Slater-Jastrow Ansatz, while the FermiNet is substantially more accurate. (b) The CO and N_2 molecules (bond lengths 2.17328 a_0 and 2.13534 a_0, respectively) with increasing numbers of determinants. All-electron CCSD(T)/CBS results are used as the baseline. No matter how many determinants are used, the FermiNet far exceeds the accuracy of the Slater-Jastrow Net.

where

$$\hat{\mathbf{a}}_\ell = \mathbf{a}_\ell - \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell], \quad \hat{\mathbf{e}}_\ell = \mathbf{e}_\ell - \mathbb{E}_{p(\mathbf{X})}[\mathbf{e}_\ell].$$

We illustrate the advantage of using KFAC over more commonly used stochastic first-order optimization methods for neural networks in Fig. 3.

III. RESULTS

Here we evaluate the performance of the FermiNet on a variety of problems in chemistry and electronic structure. Further details on the exact architectures and training procedures for the FermiNet and baselines can be found in Appendix A.

A. Slater-Jastrow versus FermiNet Ansatz

To demonstrate the expressive power of the FermiNet, we first investigated its performance relative to the more conventional Slater-Jastrow and Slater-Jastrow-backflow Ansatz with varying numbers of determinants:

$$\Psi_{\mathrm{SJ}} = e^{J(\{\mathrm{r}_i\})} \sum_k \omega_k \det\left[\phi_i^{k\uparrow}(\mathbf{r}_j^{\uparrow})\right] \det\left[\phi_i^{k\downarrow}(\mathbf{r}_j^{\downarrow})\right], \tag{13}$$

$$\Psi_{\mathrm{SJB}} = e^{J(\{\mathrm{r}_i\})} \sum_k \omega_k \det\left[\phi_i^{k\uparrow}(\mathbf{q}_j^{\uparrow})\right] \det\left[\phi_i^{k\downarrow}(\mathbf{q}_j^{\downarrow})\right], \tag{14}$$

where $\{\phi_i^{k\alpha}(\mathbf{r}_j)\}$ is a set of single-particle orbitals, typically obtained from a Hartree-Fock or density functional theory calculation, and the Jastrow factor, J is a function of the electron and nuclear coordinates. The Slater-Jastrow-backflow wave-function Ansatz replaces the electron coordinates in the orbitals with a set of collective coordinates, given by $\mathbf{q}_i = \mathbf{r}_i + \xi_i(\{\mathbf{r}_j\})$, where the backflow functions ξ_i depend on electron and nuclear coordinates and contain additional optimizable parameters.

In addition to conventional Slater-Jastrow and Slater-Jastrow backflow wave functions, we also compare against neural network versions. Rather than using Hartree-Fock orbitals, a closed-form Jastrow factor, and a backflow transform with only a few free parameters, our Slater-Jastrow-backflow network uses residual neural networks to represent the one-electron orbitals, Jastrow factor and backflow transform, making it much more flexible. The determinant part of the Slater-Jastrow network amounts to removing the two-electron stream and interactions between the one-electron streams from FermiNet. We used the conventional backflow transformation [Eq. (A4)], in which the orbitals depend on a single three-dimensional linear combination of electron position vectors and a nonlinear function of interparticle distances. Further details are provided in Appendix A 2.

To fairly compare our calculations against previous work, we first looked at single-determinant Ansatz for first-row atoms. Figure 2(a) compares the FermiNet results with numbers already available in the literature [44]. The neural network Slater-Jastrow Ansatz already outperforms the numbers from the literature by a few milli-Hartrees (mE_h), which could be due to the lack of basis set approximation error when using a neural network to represent the orbitals and a flexible Jastrow factor. The FermiNet cuts the error relative to the Slater-Jastrow Ansatz without backflow by almost an order of magnitude, and more than a factor of two relative to the Slater-Jastrow-backflow Ansatz. Just a single FermiNet determinant is sufficient to come within a few mE_h of chemical accuracy, defined as 1 kcal/mol (1.594 mE_h), which is the typical standard for a quantum chemical calculation to be considered "correct."

Not only is the FermiNet a significant improvement over the Slater-Jastrow Ansatz with one determinant, but only a few FermiNet determinants are necessary to saturate performance. Figure 2(b) shows the Slater-Jastrow network and FermiNet energies of the nitrogen and carbon monoxide molecules as functions of the number of determinants. As FCI calculations are impractical for these systems, we compare against the unrestricted coupled cluster singles, doubles, and perturbative

triples method [CCSD(T)] in the complete basis set (CBS) limit to provide a comparable baseline for both systems. As the Slater-Jastrow network optimizes all orbitals separately, the results from the Slater-Jastrow network should be a lower bound on the performance of a Slater-Jastrow Ansatz with a given number of determinants. As expected, the Slater-Jastrow network is still far from the accuracy of CCSD(T) at 64 determinants. The 64-determinant FermiNet, in contrast, comes within a few mE_h of CCSD(T). While the Slater-Jastrow-backflow Ansatz with large numbers of determinants did not completely converge, the trend is clear that the FermiNet cuts the error roughly in half. The FermiNet energies begin to plateau after only a few tens of determinants, suggesting that large linear combinations of FermiNet determinants are not required. Despite recent advances in optimal determinant selection [48,49], conventional Slater-Jastrow VMC calculations typically require tens of thousands of determinants for systems of this size and rarely match CCSD(T) accuracy even then.

B. Equilibrium geometries

Tables I and II show that the same 16-determinant FermiNet with the same training hyperparameters generalizes well to a wide variety of atoms and diatomic and small organic molecules, while Fig. 3 shows the optimization progress over time for many of these systems. As a baseline, we used a combination of experimental and exact computational results where available [34,46,47], and all-electron CBS CCSD(T) otherwise. On all atoms, as well as LiH, Li_2, methane and ammonia, the FermiNet error was within chemical accuracy. In comparison, energies from VMC using a conventional Slater-Jastrow-backflow Ansatz for first-row atoms [44] are uniformly worse than the FermiNet, despite using at least an order of magnitude more determinants. The VMC-based FermiNet energies are more comparable in quality to diffusion Monte Carlo (DMC), which is typically much more accurate than VMC. On molecules as large as ethene (C_2H_4) we recover over 99% of the correlation energy, while for larger systems like methylamine (CH_3NH_2), ozone (O_3), ethanol (C_2H_5OH) and bicyclobutane (C_4H_6) the percentage of correlation energy recovered declines gradually to ∼97%—still remarkably good for a variational calculation. Bicyclobutane is an especially challenging system due to its high ring strain and large number of electrons.

We also compare against CCSD(T) in a finite basis set in Table II, and find that in all cases the FermiNet is *more* accurate than CCSD(T) in the largest basis set we could practically run calculations on (quintuple ζ for most systems, quadruple ζ for large systems). This suggests that a comparable extrapolation of FermiNet results could match or even exceed the accuracy of CCSD(T). As the FermiNet works directly in the continuum and does not depend on a basis set, the natural equivalent would be extrapolation to the limit of infinitely wide layers in the one-electron stream. Our analysis of the FermiNet with different numbers of layers and layer widths in Sec. IV C shows that the error appears to decrease polynomially with layer width.

We also computed the first ionization potentials, $E(X^+) - E(X)$ for element X, and electron affinities, $E(X) - E(X^-)$,

TABLE I. Ground state energy, ionization potential, and electron affinity for first-row atoms. The QMC method (FermiNet, conventional VMC, or DMC) closest to the exact ground-state energy for each atom is in bold. Electron affinities for Be, N, and Ne are not computed as their anions are unstable. Experimental ionization potentials and electron affinities have had estimated relativistic effects [45] removed. All ground-state energies are within chemical accuracy of the exact numerical solution, and all electron affinities and all ionization potentials except neon are within chemical accuracy of experimental results. If no citation is provided, then the number was from our own calculation.

	Ground-state energy (E_h)							Ionization potential (mE_h)			Electron affinity (mE_h)		
Atom	FermiNet	VMC [44]	DMC [44]	CCSD(T)/CBS	HF/CBS	Exact [34]	% corr	FermiNet	Expt. [45]	ΔE	FermiNet	Expt. [45]	ΔE
Li	−7.47798(1)	−7.478034(8)	**−7.478067(5)**	−7.478157	−7.432747	−7.47806032	99.82(3)	198.10(4)	198.147	0.04(4)	21.82(20)	22.716	0.89(20)
Be	**−14.66733(3)**	−14.66719(1)	−14.667306(7)	−14.66737	−14.57301	−14.66736	99.97(3)	342.77(18)	342.593	−0.17(18)	–	–	–
B	−24.65370(3)	−24.65337(4)	**−24.65379(3)**	−24.65373	−24.53316	−24.65391	99.83(3)	304.86(4)	304.979	0.12(4)	9.03(11)	10.336	1.31(11)
C	**−37.84471(5)**	−37.84377(7)	−37.84446(6)	−37.8448	−37.6938	−37.8450	99.81(3)	413.98(8)	414.014	0.03(8)	46.18(9)	46.610	0.43(9)
N	**−54.58882(6)**	−54.5873(1)	−54.58867(8)	−54.5894	−54.4047	−54.5892	99.80(3)	534.80(12)	534.777	−0.03(12)	–	–	–
O	**−75.06655(7)**	−75.0632(2)	−75.0654(1)	−75.0678	−74.8192	−75.0673	99.70(3)	500.29(26)	500.453	0.17(26)	53.55(19)	53.993	0.44(19)
F	**−99.7329(1)**	−99.7287(2)	−99.7318(1)	−99.7348	−99.4168	−99.7339	99.69(3)	640.86(41)	640.949	0.09(41)	125.71(26)	125.959	0.25(26)
Ne	**−128.9366(1)**	−128.9347(2)	**−128.9366(1)**	−128.9394	−128.5479	−128.9376	99.74(3)	794.30(52)	794.409	0.11(52)	–	–	–

TABLE II. Ground-state energy at equilibrium geometry for diatomics and small molecules. The percentage of correlation energy captured by the FermiNet relative to the exact energy (where available) or CCSD(T)/CBS is given in the rightmost column. If no citation is provided, then the number was from our own calculation. Geometries for larger molecules are given in Appendix G.

			CCSD(T) (E_h)			HF (E_h)		
Molecule	Bond length (a_0)	FermiNet (E_h)	aug-cc-pCVQZ	aug-cc-pCV5Z	CBS	CBS	Exact (E_h)	% corr
LiH	3.015	−8.07050(1)	−8.0687	−8.0697	−8.070696	−7.98737	−8.070548 [46]	99.94(1)
Li_2	5.051	−14.99475(1)	−14.9921	−14.9936	−14.99507	−14.87155	−14.9954 [47]	99.47(1)
NH_3	–	−56.56295(8)	−56.5535	−56.5591	−56.5644	−56.2247	–	99.57(2)
CH_4	–	−40.51400(7)	−40.5067	−40.5110	−40.5150	−40.2171	–	99.66(3)
CO	2.173	−113.3218(1)	−113.3047	−113.3154	−113.3255	−112.7871	–	99.32(3)
N_2	2.068	−109.5388(1)	−109.5224	−109.5327	−109.5425	−108.9940	−109.5423 [47]	99.36(2)
Ethene	–	−78.5844(1)	−78.5733	−78.5812	−78.5888	−78.0705	–	99.16(2)
Methylamine	–	−95.8554(2)	−95.8437	–	−95.8653	−95.2628	–	98.36(3)
Ozone	–	−225.4145(3)	−225.3907	−225.4119	−225.4338	−224.3526	–	98.42(3)
Ethanol	–	−155.0308(3)	−155.0205	–	−155.0545	−154.1573	–	97.36(4)
Bicyclobutane	–	−155.9263(6)	−155.9216	–	−155.9575	−154.9372	–	96.94(5)

for first-row atoms (Table I) and compare to experimental data [45] with relativistic effects removed. Agreement with experiment is excellent (mean absolute error of 0.09 mE_h for ionization potentials and 0.66 mE_h for electron affinities), demonstrating that the FermiNet Ansatz is capable of representing charged and neutral species with similar accuracy.

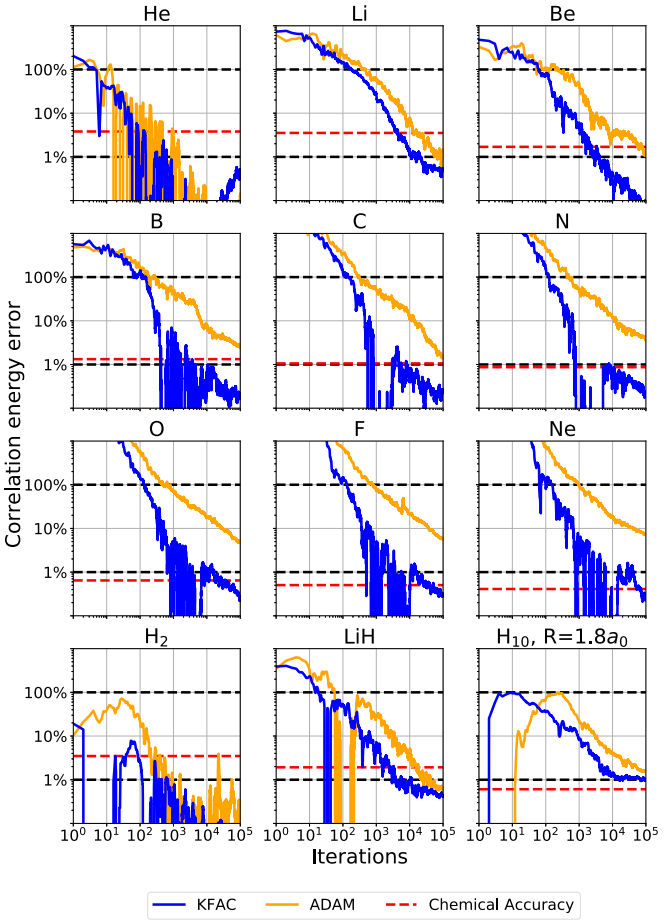

FIG. 3. Optimization progress for first-row atoms, H_2, LiH and the hydrogen chain with KFAC (blue) vs. ADAM (orange). The qualitative advantage of KFAC is clear. For clarity, the median energy over the last 10% of iterations is shown. Note that the small overshoot with KFAC between 10^3 and 10^4 iterations is due to the slow equilibration of the MCMC chain and goes away with a larger Metropolis-Hastings proposal step size.

There are many possible causes for the decline in the percent of correlation energy recovered for large systems like bicyclobutane. It may be that the FermiNet has issues with size-extensivity for larger systems. However, the FermiNet outperforms CCSD(T) in a fixed basis set, and the exponential Ansatz used by coupled cluster is size extensive, suggesting that the issue may instead be the finite width of our neural network layers. In fact, our results are with a fixed-width network, while the total number of basis functions grows with the system size for coupled cluster, meaning the coupled cluster Ansatz becomes more expressive for larger systems while the FermiNet stays fixed. Other avenues for improvement include increasing the batch size/number of walkers, improving the MCMC chain mixing and optimization efficiency, and increasing the number of determinants.

C. The H_4 rectangle

While CCSD(T) is exceptionally accurate for equilibrium geometries, it often fails for molecules with low-lying excited states or stretched, twisted or otherwise out-of-equilibrium geometries. Understanding these systems is critical for predicting many chemical properties. A model system small enough to be solved exactly by FCI for which coupled cluster fails is the rectangle of four hydrogen atoms, parametrized by the distance R of the atoms from the center and the angle θ between neighboring atoms [50]. FCI shows that the energy varies smoothly with θ and is maximized when the atoms are at the corners of a square ($\theta = 90^o$). The coupled cluster results are nonvariational, predicting energies too low by several milli-Hartree, and qualitatively incorrect, predicting an energy *minimum* with a nonanalytic downward-facing cusp at 90^o, caused by a crossing of two Hartree-Fock states with different symmetries [51]. Figure 4 shows that the FermiNet does not suffer from the same errors as coupled cluster and is in essentially perfect agreement with FCI. We attribute the small discrepancy between the FermiNet and FCI energies to

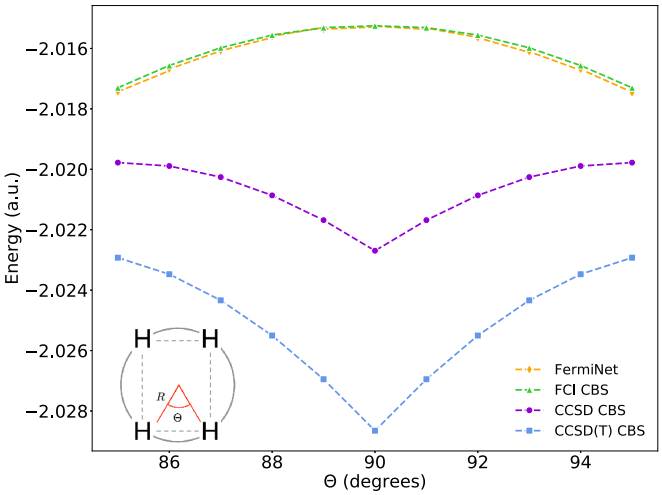

FIG. 4. The H_4 rectangle, $R = 3.2843a_0$. Coupled cluster methods incorrectly predict a cusp and energy minimum at $\Theta = 90^\circ$, while the FermiNet approach agrees with exact FCI results.

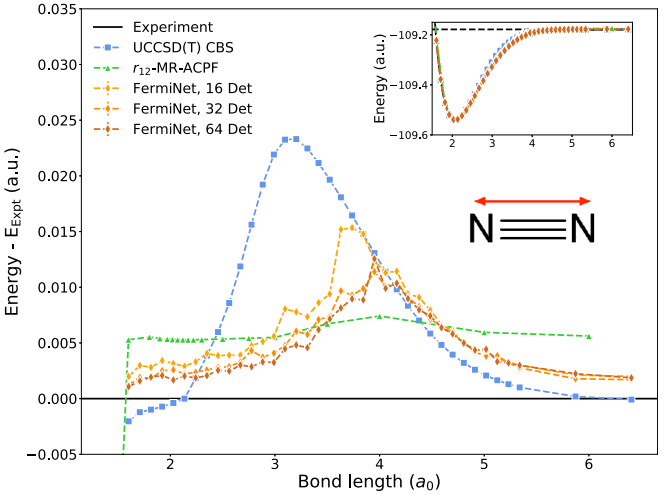

FIG. 5. The dissociation curve for the nitrogen triple-bond. The difference from experimental data [53] is given in the main panel. In the region of largest UCCSD(T) error, the FermiNet prediction is comparable to highly accurate r_{12}-MR-ACPF results [54].

errors arising from the basis set extrapolation used for the FCI energies.

D. The nitrogen molecule

A problem more relevant to real chemistry that troubles coupled cluster methods is the dissociation of the nitrogen molecule. The triple bond is challenging to describe accurately and the stretched N_2 molecule has several low-lying excited states, leading to errors when using single-reference coupled cluster methods [52]. Experimental values for the dissociation potential have been reconstructed from spectroscopic measurements using the Morse/long-range potential [53]. These closely match calculations using the r_{12}-MR-ACPF method [54], which is highly accurate but scales factorially. A comparison between unrestricted CCSD(T), the FermiNet, and these high-accuracy methods is given in Fig. 5. The total FermiNet error is significantly smaller than UCCSD(T), and in the region of largest UCCSD(T) error the FermiNet reaches accuracy comparable to r_{12}-MR-ACPF but scales much more favorably with system size. Increasing the number of determinants in the FermiNet improves performance up to a point but not beyond 32 determinants, again suggesting that the bottleneck to performance is not size-consistency. While coupled cluster could in theory be made more accurate by extending to full triples or quadruples, or using multireference methods, CCSD(T) is generally considered the largest coupled cluster approximation that can reasonably scale beyond small molecules. This shows that, without any specific tuning to the system of interest, the FermiNet is a clear improvement over single-reference coupled cluster for modeling a strongly correlated real-world chemical system.

E. The hydrogen chain

Finally, we investigated the performance of the FermiNet on the evenly spaced linear hydrogen chain. The hydrogen chain is of great interest as a system that bridges model Hamiltonians and real material systems and may undergo an insulator-to-metal transition as the separation of the atoms is decreased. Consequently, results obtained using a wide range of many-electron methods have been rigorously evaluated and compared [55]. We compare the performance of the FermiNet against many of these methods in Fig. 6. Of the two projector QMC methods studied by Motta *et al.*, AFQMC gave slightly better results than lattice regularized DMC and so we omit the latter for clarity. Without changing the network architecture or hyperparameters, we are again able to outperform coupled

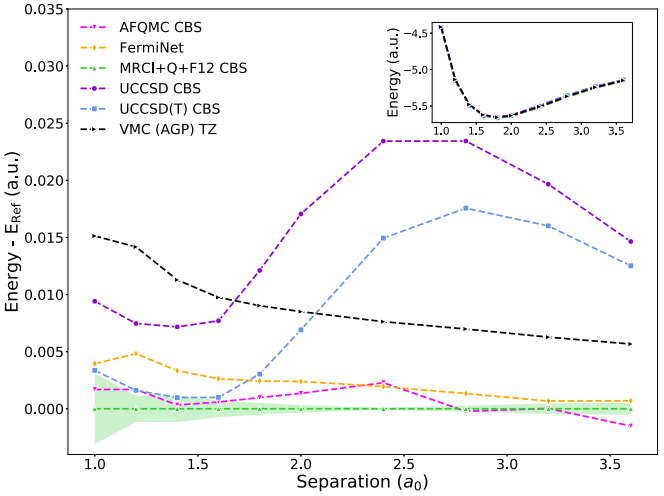

FIG. 6. The H_{10} chain. All energies except the FermiNet are taken from Motta *et al.* (2017) [55]. The absolute energies (inset) cannot be distinguished by eye. The difference from highly accurate MRCI+Q+F12 results are shown in the main panel, where the shaded region indicates an estimate of the basis-set extrapolation error. The errors in the coupled cluster and conventional VMC energies are large at medium atomic separations but the FermiNet remains comparable to AFQMC at all separations. See also Appendix E for data on larger separations.

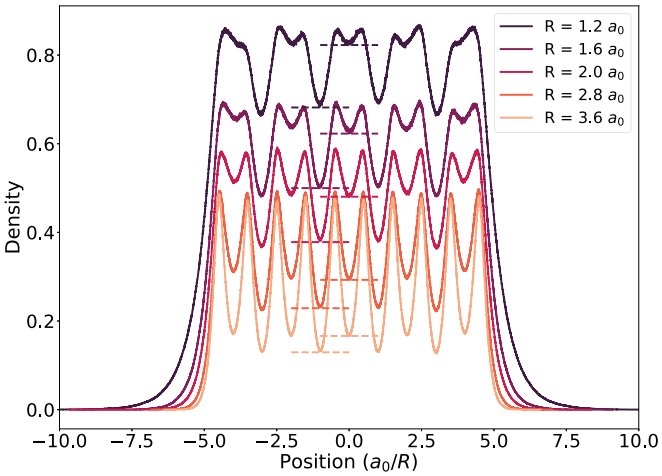

FIG. 7. Electron dimerization in the hydrogen chain. The gap between alternating minima of the density shrinks with increasing nuclear separation.

cluster methods and conventional VMC and obtain results competitive with state-of-the-art approaches.

IV. ANALYSIS

Here we provide an analysis of the performance of the FermiNet, looking at scaling with system size, network size, and visualising quantities beyond just total energy of the system.

A. Electron densities

One advantage of VMC over other *ab initio* electronic structure methods is the ease of evaluating expectation values of arbitrary observables. For instance, forces are significantly easier to calculate with VMC than projector QMC methods [56]. To illustrate the quality of the FermiNet Ansatz for observables other than energy, we analyzed the one- and two-electron densities. The electron-electron and electron-nuclear cusps for the helium atom are investigated in Appendix F.

For the hydrogen chain, we computed the one-electron density $n(\mathbf{r})$ at different nuclear separations, shown in Fig. 7. Consistent with many other electronic structure methods [57], we found that the electron density undergoes a dimerization—the density clusters around pairs of nuclei—and the effect becomes stronger with less separation between nuclei. Dimerization is a hallmark of electronic structure in insulators, and understanding when and where it occurs helps understand metal-insulator phase transitions in materials.

Additionally, we investigated the two-electron density $n(\mathbf{r}, \mathbf{r}')$ for the neon atom (Fig. 8). Understanding the behavior of the two-electron density is important for many-electronic structure methods, for instance for analyzing functionals for DFT [58]. What is interesting about the two-electron density is how it *differs* from the product of one-electron densities, $n(\mathbf{r})n(\mathbf{r}')$. This can be expressed in terms of the exchange-correlation hole, $n_{xc}(\mathbf{r}, \mathbf{r}')$, defined such that $n(\mathbf{r}, \mathbf{r}') = [n(\mathbf{r}) + n_{xc}(\mathbf{r}, \mathbf{r}')]n(\mathbf{r}')$, or in terms of the pair-correlation function, $g(\mathbf{r}, \mathbf{r}')$, defined by $n(\mathbf{r}, \mathbf{r}') = n(\mathbf{r})g(\mathbf{r}, \mathbf{r}')n(\mathbf{r}')$. As most of the density is concentrated near $\mathbf{r} = \mathbf{0}$, $n_{xc}(\mathbf{r}, \mathbf{r}')$ is very strongly peaked near $\mathbf{r} = \mathbf{0}$, obscuring its other features. We therefore show $\frac{n_{xc}(\mathbf{r},\mathbf{r}')}{n(\mathbf{r})} = 1 - g(\mathbf{r}, \mathbf{r}')$ in Fig. 8. This behaves as expected when $\mathbf{r}$ is close to $\mathbf{r}'$, showing that, at least for first- and second-order statistics, the FermiNet Ansatz is smooth and well-behaved.

B. Scaling and computation time

One of our main claims about the FermiNet is that it scales favorably compared to other *ab initio* quantum chemistry methods. The ability to run at all on systems the size of bicyclobutane proves the FermiNet scales more favorably than exact methods like FCI, but the scaling relative to other approximate methods is a more subtle question. Both the size of the FermiNet (number of hidden units, number of layers, number of determinants) and the number of training iterations required to reach a certain level of accuracy are unknown, and likely depend on the system being studied. What can be quantified is the computational complexity of a single iteration of training, which can be seen as a lower bound on the computational complexity of training the FermiNet to a certain level of accuracy.

For a system with N_e electrons, N_a atoms and a FermiNet with L hidden layers, n_1 (n_2) hidden units per one-electron (two-electron) layer and n_k determinants, evaluating the one-electron stream of the network scales as $\mathcal{O}\{N_e[N_a + L(n_1^2 + n_1 n_2)]\}$, evaluating the two-electron stream scales as $\mathcal{O}(N_e^2 L n_2^2)$, evaluating the orbitals and envelope scales as $\mathcal{O}[N_e^2 n_k(n_1 + N_a)]$, and evaluating the determinants scales as $\mathcal{O}(N_e^3 n_k)$, so the determinant calculation should dominate as N_e grows for a fixed network architecture determined by $\{L, n_1, n_2, n_k\}$. While evaluating the gradient of a function has the same asymptotic complexity as evaluating the function, evaluating the local energy scales with an additional multiplicative factor of N_e, as computing the Laplacian has the same complexity as computing the Hessian with respect to the inputs, giving an asymptotic complexity of $\mathcal{O}(N_e^4 n_k)$ as N_e grows. A Markov chain Monte Carlo (MCMC) step for sampling from ψ^2 also has the same asymptotic complexity as network evaluation for all-electron moves, or similar complexity to Laplacian calculation for single-electron moves if all electrons are moved in each loop of training.

The number of total parameters scales as $\mathcal{O}(N_a n_1 + L(n_1^2 + n_1 n_2 + n_2^2) + N_e n_k(n_1 + N_a) + n_k)$ (see Table IV for the exact shapes for each parameter). Note that, other than the orbital shaping and envelope parameters, there is no direct dependence on N_e. KFAC requires a matrix inversion for each Kronecker-factorized block of the approximate FIM, which scales as $\mathcal{O}(m^3 + n^3)$ for a linear layer with m inputs and n outputs. For the FermiNet, this works out to a scaling of $\mathcal{O}[N_a^3 n_1^3 + L(n_1^3 + n_2^3) + (N_e n_k n_1)^3 + (N_e n_k N_a)^3 + n_k^3)]$. Combining the MCMC steps, local energy calculation and KFAC update together gives an overall quartic asymptotic scaling with system size for a single step of wave-function optimization. We emphasize that the analysis here treats system size, network size and number of samping steps independently, and that the exact dependence of network size and sampling parameters on system size to achieve constant accuracy requires further investigation.

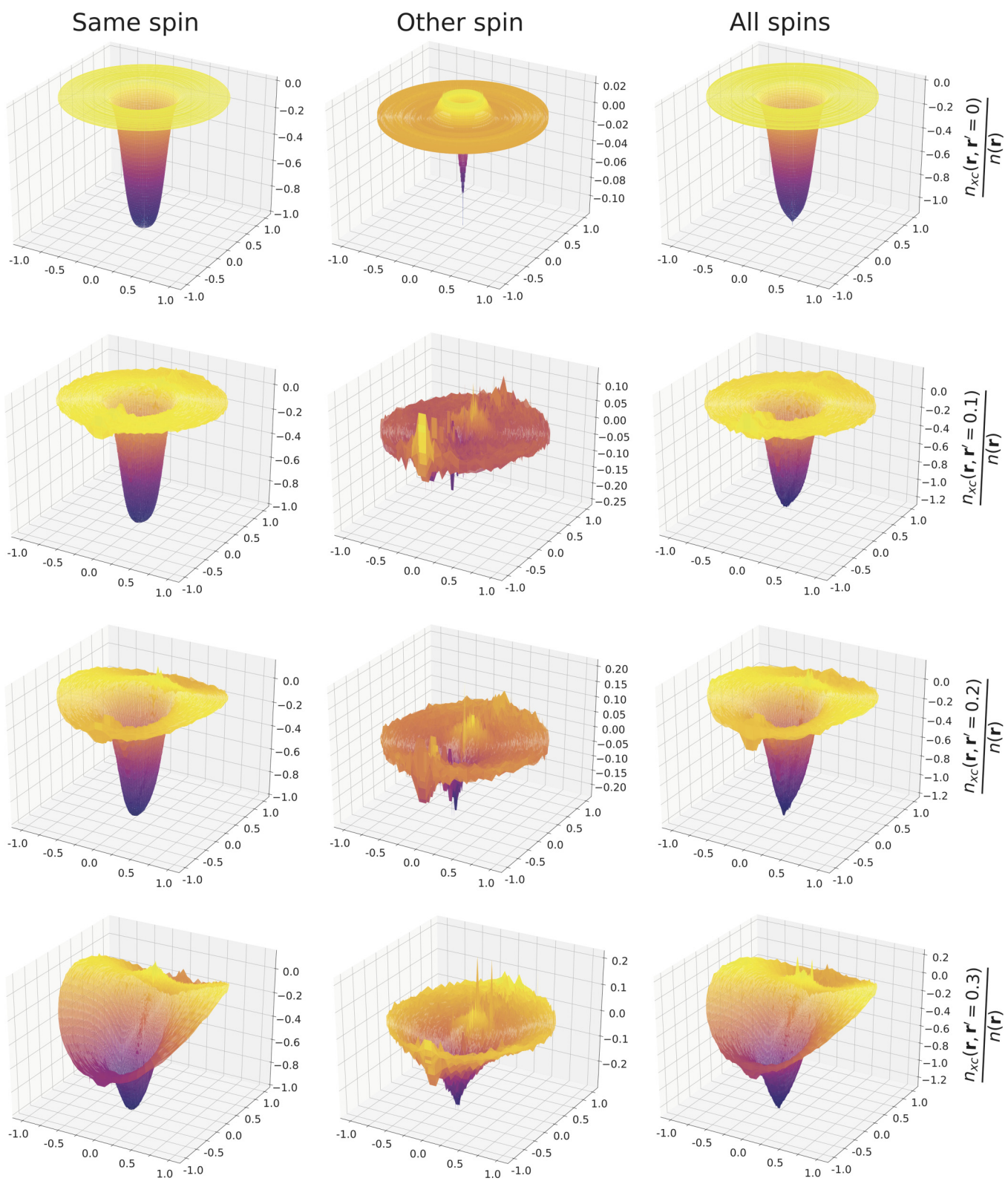

FIG. 8. The pair-correlation function $1 - g(\mathbf{r}, \mathbf{r}')$ for the neon atom, where $n(\mathbf{r}, \mathbf{r}') = n(\mathbf{r})g(\mathbf{r}, \mathbf{r}')n(\mathbf{r}')$. Different columns show the hole for electrons of the same spin (left), different spins (middle), or all electrons (right). Different rows show the hole when $\mathbf{r}'$ is between 0 and 0.3 Bohr radii from the nucleus.

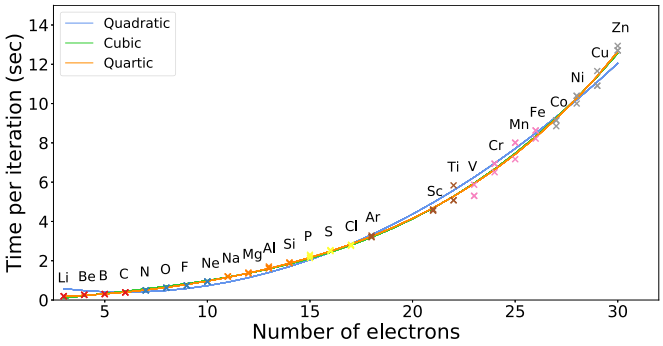

FIG. 9. Comparison of the runtime for one optimization iteration on atoms up to zinc. Polynomial regressions up to fourth order are fit to the data. The small difference between the cubic and quartic fit suggests that the determinant computation is not the dominant factor at this scale.

We give an empirical analysis of the scaling of iteration time in Fig. 9 on atoms from lithium to zinc, using the default training configuration with 8 GPUs. For larger atoms, we were not able to run optimization to convergence, but we were able to execute enough updates to get an accurate estimate of the timing for a single iteration consisting of 10 MCMC steps, a local energy and gradient evaluation and a KFAC update. Fitting polynomials of different order to the curve, we find a cubic fit is able to accurately match the scaling, suggesting that for systems of this size the computation is dominated by the $O(N^2)$ evaluation of the two-electron stream of the FermiNet, while the determinant only becomes dominant for much larger systems.

C. Feature ablation and network architectures

There are many free parameters in the FermiNet architecture that must be chosen to maximize accuracy for a given amount of computation. To illustrate the effect of different architectural choices, we removed many features, layers and hidden units from the FermiNet and investigated how the performance decayed. The FermiNet has 4 distinct input features: the nuclear coordinates $\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$ and nuclear distances $|\mathbf{r}_{iI}|$, which are inputs to the one-electron stream, and the interelectron coordinates $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and interelectron distances $|\mathbf{r}_{ij}|$, which are inputs to the two-electron stream. We compared the accuracy of the FermiNet with and without these features on the oxygen atom in Table III. All networks included the nuclear coordinates. Without the nuclear distances, the network became unstable and training crashed, possibly due to the inability to accurately capture the electron-nuclear cusp conditions. When including interelectron features, most of the increase in accuracy was due to the distances $|\mathbf{r}_{ij}|$, while the coordinates $\mathbf{r}_{ij}$ also improved accuracy, though not by as large an amount. This shows that all input features contributed towards stability and accuracy, especially the distance features. Even though a smooth neural network can approximate the nonsmooth cusps to high precision (although not perfectly), by including distances, which are nonsmooth at zero, we can make the wave function significantly easier to approximate.

TABLE III. Performance of the FermiNet on the oxygen atom with input features removed. All configurations without the electron-nuclear distances $|\mathbf{r}_{iI}|$ were numerically unstable and diverged. All numbers are relative to Chakravorty (1993) [34].

ΔE (mE_h)	Without $\mathbf{r}_{ij}$	With $\mathbf{r}_{ij}$
Without $\lvert\mathbf{r}_{ij}\rvert$	89.7	28.4
With $\lvert\mathbf{r}_{ij}\rvert$	1.2	**0.8**

To understand the effect of the size and shape of the network, we compared the FermiNet with different numbers of layers and hidden units on the hydrogen chain H_{10}. The results are presented in Fig. 10. When increasing the number of layers, the overall accuracy increases as more layers are added, but the difference from three to four layers is only on the order of 1 mE_h, suggesting that the gains from additional layers would be minor. When adding more hidden units to the one-electron stream but keeping 32 units in the two-electron stream, the accuracy increases uniformly with more units. Based on a linear regression of the log-errors relative to MRCI+Q+F12, and using bootstrapping to generate error bars, the error scales with the number of hidden units in the one-electron stream as $\mathcal{O}(N^{-0.395\pm0.067})$. This means we would expect around 760 hidden units to be needed to reach chemical accuracy on the hydrogen chain. For the two-electron stream, the improvement with more units quickly saturates. In fact, going from 16 to 32 hidden units seems to make the results slightly noisier. This suggests that increasing the width of the one-electron stream, more than increasing the width of the two-electron stream or the total depth, is the most promising route to increasing overall accuracy of the FermiNet.

V. DISCUSSION

We have shown that antisymmetric neural networks can be constructed and optimized to enable high-accuracy quantum chemistry calculations of challenging systems. The Fermionic Neural Network makes the simple and straightforward VMC method competitive with DMC, AFQMC, and CCSD(T) methods for equilibrium geometries and better than CCSD(T) for many out-of-equilibrium geometries. Importantly, one network architecture with one set of training parameters has been able to attain high accuracy on every system examined. The use of neural networks means that we do not have to choose a basis set or worry about basis-set extrapolation, a common source of error in computational quantum chemistry. There are many possible applications of the FermiNet beyond VMC, for instance as a trial wave function for projector QMC methods. We expect further work investigating the tradeoffs of different antisymmetric neural networks and optimization algorithms to lead to greater computational efficiency, higher representational capacity, and improved accuracy on larger systems. This has the potential to bring to quantum chemistry the same rapid progress that deep learning has enabled in numerous fields of artificial intelligence.

ACKNOWLEDGMENTS

We thank J. Jumper, J. Kirkpatrick, M. Hutter, T. Green, N. Blunt, S. Mohamed, and A. Cohen for helpful discussions, B.

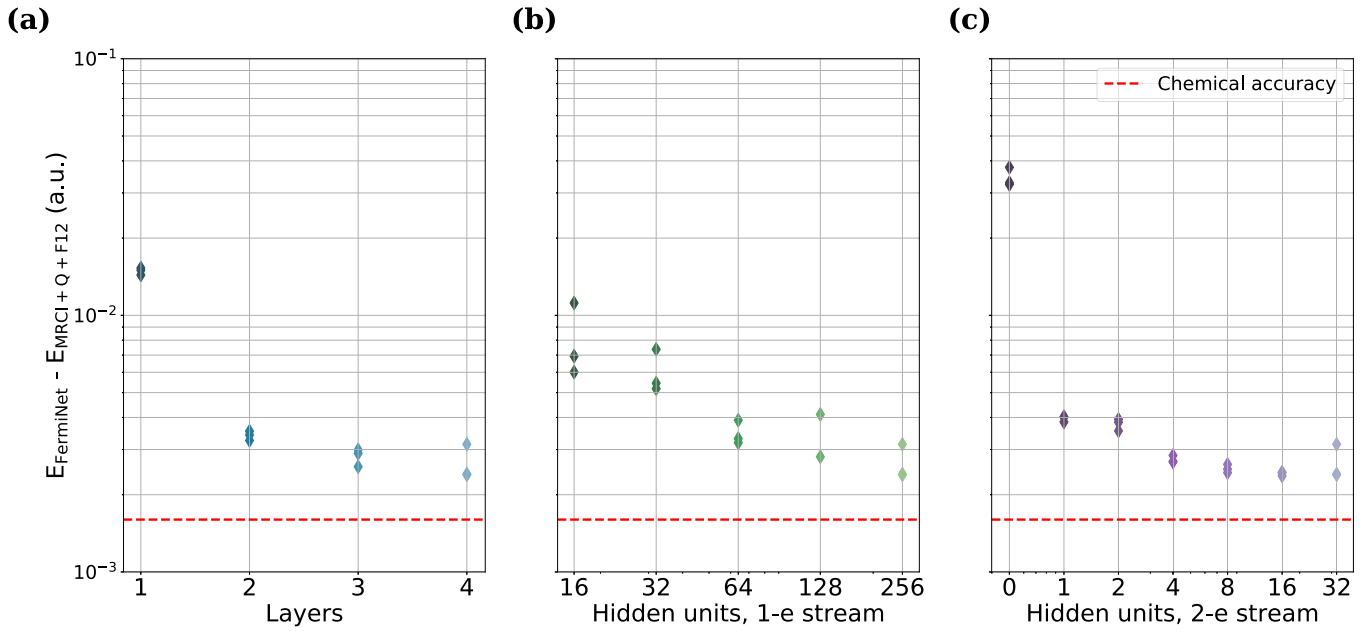

FIG. 10. Effects of network architecture on FermiNet performance on the hydrogen chain H_{10} with separation $R = 2.0a_0$. Each point is one run of the same model. (a) Effect of network depth. The marginal improvement with 4 layers is small but not zero. (b) Effect of number of hidden units in the one-electron stream. There is a continuous improvement with wider layers, with the error decreasing roughly as $\mathcal{O}(N^{-0.395\pm0.067})$. (c) Effect of number of hidden units in the two-electron stream. The accuracy plateaus above 16 units.

McMorrow for providing data, J. Martens and P. Buchlovsky for assistance with code, and A. Obika, S. Nelson, C. Meyer, T. Back, S. Petersen, P. Kohli, K. Kavukcuoglu, and D. Hassabis for support and guidance. Additional thanks to the rest of the DeepMind team for support, ideas, and encouragement.

APPENDIX A: EXPERIMENTAL SETUP

1. FermiNet architecture and training

For all experiments, a Fermionic neural network with four layers was used, not counting the final linear layer that outputs the orbitals. Each layer had 256 hidden units for the one-electron stream and 32 hidden units for the two electron stream. A tanh nonlinearity was used for all layers, as a smooth function is needed to guarantee that the Laplacian is well defined and nonzero everywhere. 16 determinants were used where not otherwise specified. For comparison, the conventional VMC results in Table I from Seth *et al.* (2011) [44] use 50 configuration state functions (CSF). While the exact number of determinants in a CSF will depend on the system, generally this will be on the order of hundreds to thousands of determinants. With this configuration of the FermiNet there were approximately 700 000 parameters in the network, although the exact number depends on the number of atoms in the system due to the way we construct the input features and exponentially decaying envelope. A breakdown of these parameters are given in Table IV.

Before using the local energy as an optimization objective we pretrained the network to match Hartree-Fock (HF) orbitals computed using PySCF [59]. There were two reasons for this. First, we found that the numerical stability of the subsequent local energy optimization was improved. On large systems, the determinants in the Fermionic neural network would often numerically underflow if no pretraining was used, causing the optimization to fail. Pretraining with HF orbitals as a guide meant that the main optimization started in a region of relatively low variance, with comparatively stable determinant evaluations and electron walkers in representative configurations. Second, we found that time was saved by not optimizing the local energy through a region that we knew to be physically uninteresting, given that it had an energy higher than that of a straightforward mean-field approximation. The pretraining did not seem to strand the neural network in a poor local optimum, as the energy minimization always gave consistent results capturing roughly 99% of the correlation energy. This is consistent with the conventional wisdom in the machine learning community that issues with local minima are less severe in wider, deeper neural networks. Further, stochasticity in the optimization procedure helps break symmetry and escape bad minima.

The pretraining loss is

$$\mathcal{L}^{\text{pre}}(\theta) = \int \left\{ \sum_{\alpha\in\{\uparrow,\downarrow\}} \sum_{ijk} \left[\phi_i^{k\alpha}\left(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar{\alpha}}\}\right) - \phi_{i\alpha}^{\text{HF}}\left(\mathbf{r}_j^\alpha\right) \right]^2 \right\} p^{\text{pre}}(\mathbf{X}) d\mathbf{X},$$

where $\phi_{i\alpha}^{\text{HF}}(\mathbf{r}_j^\alpha)$ denotes the value of the ith Hartree-Fock orbital for spin α at the position of electron j, $\bar{\alpha}$ is $\downarrow$ if α is $\uparrow$ or vice versa, and $\phi_i^{k\alpha}(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar{\alpha}}\})$ is the corresponding entry in the input to the kth determinant of the Fermionic neural network. We use a minimal (STO-3G) basis set for the Hartree-Fock computation as we require only a stable initialization in the rough vicinity of the mean-field

TABLE IV. Network activations and parameters for FermiNet with L layers, n_k many-electron determinants for a system of N_a atoms and N_e electrons. i, j index electrons in spin channels $\alpha, \beta \in \{\uparrow, \downarrow\}$. Each layer contains n_1^ℓ (n_2^ℓ) hidden units for the one-electron (two-electron) stream. The quantity column shows the total number of each object.

Symbol	Dimension	Quantity	Learnable	Description
$\mathbf{h}_i^{0\alpha}$	$4N_a$	N_e		one-electron features
$\mathbf{h}_{ij}^{0\alpha\beta}$	4	N_e^2		two-electron features
$\mathbf{h}_i^{\ell\alpha}$	$n_1^{\ell-1}$	$(L-1)N_e$		one-electron activations from layer $\ell-1$
$\mathbf{h}_{ij}^{\ell\alpha\beta}$	$n_2^{\ell-1}$	$(L-1)N_e^2$		two-electron activations from layer $\ell-1$
$\mathbf{f}_i^{\ell\alpha}$	$3n_1^{\ell-1}+2n_2^{\ell-1}$	LN_e		one-electron input for layer ℓ
$\mathbf{V}^\ell$	$n_1^\ell \times (3n_1^{\ell-1}+2n_2^{\ell-1})$	L	✓	weights for one-electron linear layer
$\mathbf{b}^\ell$	n_1^ℓ	L	✓	biases for one-electron linear layer
$\mathbf{W}^\ell$	$n_2^\ell \times n_2^{\ell-1}$	L	✓	weights for two-electron linear layer
$\mathbf{c}^\ell$	n_2^ℓ	L	✓	biases for two-electron linear layer
$\mathbf{w}_i^{k\alpha}$	n_1^L	$n_k N_e$	✓	weights for final linear layer (orbital shaping)
$g_i^{k\alpha}$	scalar	$n_k N_e$	✓	bias for final linear layer (orbital shaping)
$\pi_{im}^{k\alpha}$	scalar	$n_k N_a N_e$	✓	enevelope weight
$\Sigma_{im}^{k\alpha}$	3×3	$n_k N_a N_e$	✓	enevelope decay
ω	n_k	1	✓	weights in determinant expansion

solution, not an accurate mean-field result. The probability distribution $p^{\text{pre}}(\mathbf{X})$ is an equal mixture of the product of Hartree-Fock orbitals and the output of the Fermionic neural network:

$$p^{\text{pre}}(\mathbf{X}) = \frac{1}{2}\left\{\prod_{\alpha\in\{\uparrow,\downarrow\}}\prod_i \left[\phi_{i\alpha}^{\text{HF}}\left(\mathbf{r}_i^\alpha\right)\right]^2 + \psi^2(\mathbf{X})\right\}.$$

We chose not to use the distribution from the Hartree-Fock determinant because we wanted sample coverage at every point where the orbitals were large, but in practice the difference to using the antisymmetrized distribution was marginal. The inclusion of the neural network density helps to increase the sampling probability in areas where the neural network wave function is spuriously high. We approximate the expectation for the loss by using MCMC to draw half the samples in the batch from ψ^2 and half from the product of Hartree-Fock orbitals using MCMC.

Initial MCMC configurations were drawn from Gaussian distributions centered on each atom in the molecule. Electrons were assigned to atoms according to the nuclear charge and spin polarization of the ground state of the isolated atom, with the atomic spins orientated such that the total spin projection of the molecule was correct, which was possible for systems studied here. We used ADAM with default parameters as the optimizer. After pretraining, we reinitialized the electron walker positions and then had a burn in MCMC period with target distribution ψ^2 before we began local energy minimization.

For the FermiNet, all code was implemented in TensorFlow 1 built with CUDA 9. All experiments for systems with less than 20 electrons were run in parallel on 8 V100 GPUs, while 16 GPUs were used for larger systems. With a smaller batch size we were able to train on a single GPU but convergence was significantly and disproportionately slower. For instance, ethene converged after just 2 days of training with 8 GPUs, while several weeks were required on a single GPU. Bicyclobutane, with 30 electrons, took roughly 1 month on 16 GPUs to train. We expect further engineering improvements will reduce this number. Ten Metropolis-Hastings steps were taken between every parameter update, and it typically required $O(10^5$–$10^6)$ parameter updates to reach convergence (results in the paper used 2×10^5 parameter updates). Conventional VMC wave-function optimization will perform $O(10^1$–$10^2)$ parameter updates and $O(10^4$–$10^6)$ MCMC steps between updates, so we require roughly the same number of wave-function evaluations as conventional VMC. After network optimization, we run $O(10^5)$ MCMC steps and calculate the mean local energy every 10 steps. The energy and associated standard error are estimated using a standard approach to account for correlations [60].

Accurate and stable convergence was highly dependent on the hyperparameters used; the default values for all experiments are included in Table V. These hyperparameters do

TABLE V. Default hyperparameters for all experiments in the paper. For bicyclobutane, the batch size was halved and the pretraining iterations were increased by an order of magnitude.

	Parameter	Value
	Batch size	4096
	Training iterations	2e5
	Pretraining iterations	1e3
	Learning rate	$(1e4+t)^{-1}$
	Local energy clipping	5.0
KFAC	Momentum	0
KFAC	Covariance moving average decay	0.95
KFAC	Norm constraint	1e-3
KFAC	Damping	1e-3
MCMC	Proposal std dev (per dimension)	0.02
MCMC	Steps between parameter updates	10

seem to be generalizable—we have observed good performance on every system investigated. For some larger systems, stability was improved by using more pretraining iterations. Getting good performance from KFAC requires careful tuning, and we found that the damping and norm constraint parameters critically affect the asymptotic performance. If the damping is too high, then KFAC behaves like gradient descent near a local minimum and converges too slowly. If the damping is reduced, then training quickly becomes unstable unless the norm constraint (a generalization of gradient clipping) is lowered in tandem. Surprisingly, we found little advantage to using momentum, and sometimes it even seemed to reduce training performance, so we set it to zero for all experiments.

To reduce the variance in the parameter updates, we clipped the local energy when computing the gradients but not when evaluating the total energy of the system. This is a commonly used strategy to improve the accuracy of QMC [61]. We computed the total variation of each batch, $\frac{1}{N}\sum_i |E_L(\mathbf{X}_i) - \tilde{E}_L|$, where $\tilde{E}_L$ is the median local energy of that batch. This is to the ℓ_1 norm what the standard deviation is to the ℓ_2 norm, and we prefer it to the standard deviation as it is more robust to outliers. We clip any local energies more than five times further from the median than this total variation and compute the gradient in Eq. (10) with the clipped energy in place of E_L. The aforementioned KFAC norm constraint enforces gradient clipping in a manner which respects the information geometry of the model.

To sample from $\psi^2(\mathbf{X})$ we used the standard Metropolis-Hastings algorithm [12]. The proposed moves were Gaussian distributed with a fixed, isotropic covariance. All electron positions were updated simultaneously. While one-electron moves are more common in VMC, prior work suggests that all-electron moves are effective at the scale of system we investigated [62] and the fact that our orbitals depend on all electrons means that we cannot exploit fast determinant updates with one-electron moves. We expect one-electron moves will have a more noticeable impact for larger systems and will investigate different MCMC strategies and parameters in future work. Typical acceptance rates were ∼0.95 for the smallest systems and ∼0.6 for the largest systems investigated. Due to slow equilibration of the MCMC sampling, the computed energy sometimes overshot the true value, but always reequilibrated after a few thousand iterations. We experimented with Hamiltonian Monte Carlo to give faster mixing and lower bias in the gradients, but found this led to significantly higher variance in the local energy and lower overall performance.

2. Slater-Jastrow networks

For the baseline Slater-Jastrow network, an multilayer perceptrons (MLP) with three hidden layers of 128 units were used for the orbitals. The electron positions and electron-nuclear vectors and distances were used as input features. The output of the MLP was fed into a final linear layer to generate the required orbitals and the same multiplicative envelope employed in the Fermionic neural network was included; this can be seen as an extension to the electron-nuclear Jastrow factor. The Jastrow factor and backflow transform are of the standard form [63]:

$$\begin{aligned} J(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\},\{\mathbf{R}\}) &= J^{(e-n)}(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\},\{\mathbf{R}\}) \\ &\quad + J^{(e-e)}(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\}) \\ &\quad + J^{(e-e-n)}(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\},\{\mathbf{R}\}), \end{aligned} \tag{A1}$$

$$J^{(e-n)}(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\},\{\mathbf{R}\}) = \sum_{\alpha\in\{\uparrow,\downarrow\}}\sum_{i=1}^{n^\alpha}\sum_{I}^{N_a}\chi_j(|\mathbf{r}_i^\alpha - \mathbf{R}_I|),$$

$$J^{(e-e)}(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\}) = \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{i=1}^{n^\alpha}\sum_{j=1}^{n^\beta}u^{\alpha\beta}\left(\left|\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta\right|\right),$$

$$\begin{aligned} J^{(e-e-n)}(\{\mathbf{r}^\uparrow\},\{\mathbf{r}^\downarrow\},\{\mathbf{R}\}) &= \sum_{\alpha,\beta\in\{\uparrow,\downarrow\}}\sum_{i=1}^{n^\alpha}\sum_{j=1}^{n^\beta}\sum_{I}^{N_a}, \\ & f_k^{\alpha\beta}\left(\left|\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta\right|, \left|\mathbf{r}_i^\alpha - \mathbf{R}_I\right|, \left|\mathbf{r}_j^\beta - \mathbf{R}_I\right|\right), \end{aligned} \tag{A2}$$

for the Jastrow factor, and

$$\begin{aligned} \mathbf{r}_i' &= \mathbf{r}_i + \xi_i^{(e-e)}(\{\mathbf{r}_j\}) + \xi_i^{(e-N)}(\{\mathbf{R}_I\}) \\ &\quad + \xi_i^{(e-e-N)}(\{\mathbf{r}_j\},\{\mathbf{R}_I\}), \end{aligned} \tag{A3}$$

$$\xi_i^{(e-e)}(\{\mathbf{r}_j\}) = \sum_{j\neq i}^{n}\eta(|\mathbf{r}_{ij}|)\mathbf{r}_{ij},$$

$$\xi_i^{(e-N)}(\{\mathbf{R}_I\}) = \sum_{I}^{N_a}\mu(|\mathbf{r}_{iI}|)\mathbf{r}_{iI},$$

$$\begin{aligned} \xi_i^{(e-e-N)}(\{\mathbf{r}_j\},\{\mathbf{R}_I\}) &= \sum_{j\neq i}^{n}\sum_{I}^{N_a}\Phi(|\mathbf{r}_{ij}|,|\mathbf{r}_{iI}|,|\mathbf{r}_{jI}|)\mathbf{r}_{ij} \\ &\quad + \Theta(|\mathbf{r}_{ij}|,|\mathbf{r}_{iI}|,|\mathbf{r}_{jI}|)\mathbf{r}_{iI}, \end{aligned} \tag{A4}$$

for the backflow transform, where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$. Here $\{\chi_j\}$, $\{u^{\alpha\beta}\}$, $\{f_k^{\alpha\beta}\}$, η, μ, Φ and Θ are all separate three-layer perceptrons with 64 hidden units. Residual connections were used in all MLPs, which greatly improved the stability of training. We found Slater-Jastrow-backflow networks to be extremely unstable to train from random initial weights and hence used a fine-tuning approach where the Slater-Jastow-backflow networks were initialized from an optimized Slater-Jastrow network with the weights and biases in the backflow MLPs randomly initialized close to zero. The Slater-Jastrow and Slater-Jastrow-backflow networks were otherwise optimized in the identical fashion to FermiNet.

3. Hartree-Fock and coupled cluster calculations

We used PySCF [59] to perform all-electron CCSD(T) calculations on atoms and dimers (Table I). PSI4 [64] was used to perform all-electron CCSD(T) calculations on all other molecules, and and FCI calculations on H_4. Cholesky decomposition [65] was used to reduce the memory requirements for bicyclobutane, which we verified introduces an error in the total energies of $\mathcal{O}(10^{-5})$ Hartrees with the aug-cc-pCVTZ basis set. The H_4 calculations used a cc-pVXZ (X = T, Q, 5) basis set. All other CCSD(T) calculations used aug-cc-pCVXZ (X = T, Q, 5) basis sets. An unrestricted

Hartree-Fock reference was used for atoms and dimers, with restricted Hartree-Fock used otherwise. We extrapolated energies to the CBS limit using standard methods [66,67]. CBS Hartree-Fock energies for Li, Be, and Li_2 were taken from aug-cc-pCV5Z calculations, in which the basis set error was below 10^{-4} Hartrees. CBS Hartree-Fock energies for other systems were obtained by fitting the function $E_{\mathrm{HF}}(X) = E_{\mathrm{HF}}(\mathrm{CBS}) + ae^{-bX}$, where X is the cardinality of the basis; CCSD, CCSD(T) and FCI correlation energies were extrapolated to the CBS by fitting the energies from quadruple- and quintuple-zeta basis sets (triple- and quadruple-ζ for bicyclobutane) to the function $E_c(X) = E_c(\mathrm{HF}) + aX^{-3}$. The total energy is given by the sum of the Hartree-Fock energy and correlation energy. To compare the dissociation potential of N_2 against experiment, we used the $\mathrm{MLR}_4(6, 8)$ potential from Le Roy *et al.* (2006) [53] which is based on fitting 19 lines of the N_2 vibrational spectrum.

APPENDIX B: UNIVERSALITY OF GENERALIZED SLATER DETERMINANTS

Empirically, the accuracy of the FermiNet increases as the number of determinants grows. This raises the question: In theory, how many determinants are necessary to represent any antisymmetric function $\psi(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ when the elements of the determinant are permutation-equivariant functions of the form $\mathbf{\Phi}_{ij} = \phi_i(\mathbf{x}_j; \{\mathbf{x}_{/j}\})$? The answer, perhaps surprisingly, is just one. The argument below is originally due to M. Hutter (personal communication).

Define a unique ordering on the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, for instance, $\mathbf{x}_i < \mathbf{x}_j$ if the first coordinate of $\mathbf{x}_i$ is less than the first coordinate of $\mathbf{x}_j$. Let π be the permutation such that $\mathbf{x}_{\pi(1)} \leqslant \mathbf{x}_{\pi(2)} \leqslant \ldots \leqslant \mathbf{x}_{\pi(n)}$, that is, π sorts the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and let $\sigma(\pi)$ be the sign of the permutation π. Define $\phi_1(\mathbf{x}_j; \{\mathbf{x}_{/j}\}) = \mathbb{1}_{j=\pi(1)}\psi(\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(n)})$ and $\phi_i(\mathbf{x}_j; \{\mathbf{x}_{/j}\}) = \mathbb{1}_{j=\pi(i)}$ if $i \neq 1$. Then each row of the matrix has only one nonzero entry, and the determinant $\det[\mathbf{\Phi}_{ij}] = \sigma(\pi)\psi(\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(n)}) = \psi(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

The functions ϕ_i are not everywhere continuous, due to the indicator functions $\mathbb{1}_{j=\pi(i)}$, and therefore not learnable by the FermiNet. This may partially explain why, despite the theoretical universality of a single determinant, in practice we still require multiple determinants to achieve high accuracy. We should note that this construction is very similar to the suggestion in Luo and Clark [21] that neural network backflow could be extended to continuous spaces by sorting the input vectors and multiplying a neural network Ansatz by the sign of the permutation. As the choice of ordering breaks a natural symmetry of the system, and the Ansatz becomes nonsmooth anywhere the ordering changes, we suspect such an Ansatz would be less effective than the FermiNet; however, it is appealingly simple.

APPENDIX C: EQUIVALENCE OF NATURAL GRADIENT DESCENT AND STOCHASTIC RECONFIGURATION

Here we provide a derivation illustrating that stochastic reconfiguration is equivalent to natural gradient descent for unnormalized distributions. Though many authors have investigated extensions of the Fisher information metric to quantum systems [68], this particular connection between methods in machine learning and quantum chemistry seems not to be widely appreciated by either community, though it was pointed out in Nomura *et al.* (2017) [19].

We denote the density proportional to $\psi^2(\mathbf{X})$ by $p(\mathbf{X})$, and the normalizing factor by $Z(\theta)$. In addition, let $\tilde{p}(\mathbf{X}) = \psi^2(\mathbf{X})$ denote the unnormalized density. In stochastic reconfiguration, the entries of the preconditioner matrix $\mathcal{M}$ have the form

$$\mathcal{M}_{ij} = \mathbb{E}_{p(\mathbf{X})}[(\mathcal{O}_i - \mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_i])(\mathcal{O}_j - \mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_j])],$$

where

$$\mathcal{O}_i(\mathbf{X}) = \psi(\mathbf{X})^{-1}\frac{\partial\psi(\mathbf{X})}{\partial\theta_i} = \frac{\partial\log|\psi(\mathbf{X})|}{\partial\theta_i} = \frac{1}{2}\frac{\partial\log\tilde{p}(\mathbf{X})}{\partial\theta_i}$$

and $\mathcal{M}$ is a metric for the parameter space [69]. The term $\mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_i]$ can be expressed in terms of the normalizing factor:

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_i] &= \frac{1}{2}\mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial\log\tilde{p}(\mathbf{X})}{\partial\theta_i}\right] \\
&= \frac{1}{2}\int\frac{\partial\log\tilde{p}(\mathbf{X})}{\partial\theta_i}p(\mathbf{X})d\mathbf{X} \\
&= \frac{1}{2}\int\frac{\partial\log\tilde{p}(\mathbf{X})}{\partial\theta_i}\frac{\tilde{p}(\mathbf{X})}{Z(\theta)}d\mathbf{X} \\
&= \frac{1}{2}\int\frac{1}{\tilde{p}(\mathbf{X})}\frac{\partial\tilde{p}(\mathbf{X})}{\partial\theta_i}\frac{\tilde{p}(\mathbf{X})}{Z(\theta)}d\mathbf{X} \\
&= \frac{1}{2Z(\theta)}\int\frac{\partial\tilde{p}(\mathbf{X})}{\partial\theta_i}d\mathbf{X} \\
&= \frac{1}{2Z(\theta)}\frac{\partial}{\partial\theta_i}\int\tilde{p}(\mathbf{X})d\mathbf{X} \\
&= \frac{1}{2Z(\theta)}\frac{\partial Z(\theta)}{\partial\theta_i} \\
&= \frac{1}{2}\frac{\partial\log Z(\theta)}{\partial\theta_i}.
\end{aligned}$$

Plugging this into the expression for $\mathcal{M}_{ij}$ yields

$$\begin{aligned}
\mathcal{M}_{ij} &= \mathbb{E}_{p(\mathbf{X})}[(\mathcal{O}_i - \mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_i])(\mathcal{O}_j - \mathbb{E}_{p(\mathbf{X})}[\mathcal{O}_j])] \\
&= \frac{1}{4}\mathbb{E}_{p(\mathbf{X})}\left[\left(\frac{\partial\log\tilde{p}(\mathbf{X})}{\partial\theta_i} - \frac{\partial\log Z(\theta)}{\partial\theta_i}\right)\right. \\
&\quad \left.\times\left(\frac{\partial\log\tilde{p}(\mathbf{X})}{\partial\theta_j} - \frac{\partial\log Z(\theta)}{\partial\theta_j}\right)\right] \\
&= \frac{1}{4}\mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial\log p(\mathbf{X})}{\partial\theta_i}\frac{\partial\log p(\mathbf{X})}{\partial\theta_j}\right],
\end{aligned}$$

which, up to a constant, is the Fisher information metric for $p(\mathbf{X})$.

APPENDIX D: NUMERICALLY STABLE COMPUTATION OF THE LOG DETERMINANT AND DERIVATIVES

For numerical stability, the Fermionic neural network outputs the *logarithm* of the absolute value of the wave function (along with its sign), and we compute log determinants rather than determinants. Even if some of the matrices are singular, this is not an issue for numerical stability on the forward pass, because these matrices will have zero contribution to the

overall sum of determinants the network outputs:

$$\log|\psi(\mathbf{r}_1^{\uparrow},\ldots,\mathbf{r}_{n^{\downarrow}}^{\downarrow})| = \log\left|\sum_k \omega_k \det[\mathbf{\Phi}^{k\uparrow}]\det[\mathbf{\Phi}^{k\downarrow}]\right|.$$

We use the "log-sum-exp trick" to compute the sum—that is, we subtract off the largest log determinant before exponentiating and computing the weighted sum, and add it back in after the logarithm at the end. This avoids numerical underflow if the log determinants are not well scaled.

Naively applying automatic differentiation frameworks to compute the gradient and Laplacian of the log wave function will not work if one of the matrices is singular. However, the first and second derivatives are still well defined, and we show how to express these derivatives in closed form appropriate for reverse-mode automatic differentiation. Several of the results used here, as well as the notation, are based on the collected matrix derivative results of Giles (2008) [70].

From Jacobi's formula, the gradient of the determinant of a matrix is given by

$$\frac{\partial \det(\mathbf{A})}{\partial \mathbf{A}} = \det(\mathbf{A})\mathbf{A}^{-T} = \mathrm{Adj}(\mathbf{A})^T = \mathrm{Cof}(\mathbf{A}),$$

where $\mathrm{Cof}(\mathbf{A})$ is the cofactor matrix of $\mathbf{A}$. Let $\mathbf{C} = \mathrm{Cof}(\mathbf{A})$. Then, by the product rule, we can express the reverse-mode gradient of $\mathrm{Cof}(\mathbf{A})$ as

$$\bar{\mathbf{A}} = \mathbf{A}^{-T}[\mathrm{Tr}(\bar{\mathbf{C}}^T\mathrm{Cof}(\mathbf{A}))\mathbf{I} - \bar{\mathbf{C}}^T\mathrm{Cof}(\mathbf{A})],$$

where $\bar{\mathbf{C}}$ is the reverse-mode sensitivity. Unfortunately, this expression becomes undefined if the matrix $\mathbf{A}$ is singular. Even so, both the cofactor matrix and its derivative are still well defined. To see this, we express the cofactor in terms of the singular value decomposition of $\mathbf{A}$. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{A}$, then

$$\begin{aligned}\mathrm{Cof}(\mathbf{A}) &= \det(\mathbf{A})\mathbf{A}^{-T}\\ &= \det(\mathbf{U})\det(\mathbf{\Sigma})\det(\mathbf{V})\mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{V}^T.\end{aligned}$$

Since $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices, their determinant is just the sign of their determinant. To avoid clutter, we drop the $\det(\mathbf{U})$ and $\det(\mathbf{V})$ terms until the very end. Let σ_i be the ith diagonal element of $\mathbf{\Sigma}$, then we have $\det(\mathbf{\Sigma}) = \prod_i \sigma_i$, and canceling terms in the expression, we get (up to a sign factor)

$$\mathrm{Cof}(\mathbf{A}) = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^T,$$

where $\mathbf{\Gamma}$ is a diagonal matrix with elements γ_i defined as

$$\gamma_i = \prod_{j\neq i}\sigma_j,$$

because the σ_i^{-1} term in $\mathbf{\Sigma}^{-1}$ cancels out one term in $\det(\mathbf{\Sigma})$.

The gradient of the cofactor is more complicated, but once again terms cancel. Again neglecting a sign factor, the reverse-mode gradient can be expanded in terms of the singular vectors as

$$\begin{aligned}\bar{\mathbf{A}} &= \mathbf{A}^{-T}[\mathrm{Tr}(\bar{\mathbf{C}}^T\mathrm{Cof}(\mathbf{A}))\mathbf{I} - \bar{\mathbf{C}}^T\mathrm{Cof}(\mathbf{A})]\\ &= \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{V}^T[\mathrm{Tr}(\bar{\mathbf{C}}^T\mathbf{U}\mathbf{\Gamma}\mathbf{V}^T)\mathbf{I} - \bar{\mathbf{C}}^T\mathbf{U}\mathbf{\Gamma}\mathbf{V}^T]\\ &= \mathbf{U}[\mathrm{Tr}(\bar{\mathbf{C}}^T\mathbf{U}\mathbf{\Gamma}\mathbf{V}^T)\mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\mathbf{V}^T\bar{\mathbf{C}}^T\mathbf{U}\mathbf{\Gamma}]\mathbf{V}^T\\ &= \mathbf{U}[\mathrm{Tr}(\mathbf{M}\mathbf{\Gamma})\mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\mathbf{M}\mathbf{\Gamma}]\mathbf{V}^T,\end{aligned}$$

where $\mathbf{M} = \mathbf{V}^T\bar{\mathbf{C}}^T\mathbf{U}$, and we have taken advantage of the invariance of the trace of matrix products to cyclic permutation in the last line.

Now, in the expression inside the square brackets in the last line, terms conveniently cancel that prevent the expression from becoming undefined should $\sigma_i = 0$ for some singular value. Denote this term $\mathbf{\Xi}$, the off-diagonal terms of $\mathbf{\Xi}$ only depend on the second term $\mathbf{\Sigma}^{-1}\mathbf{M}\mathbf{\Gamma}$:

$$\begin{aligned}\Xi_{ij} &= -M_{ij}\sigma_i^{-1}\gamma_j = -M_{ij}\sigma_i^{-1}\prod_{k\neq j}\sigma_k\\ &= -M_{ij}\prod_{k\neq i,j}\sigma_k,\end{aligned}$$

and the diagonal terms have the form

$$\begin{aligned}\Xi_{ii} &= \sigma_i^{-1}\sum_j M_{jj}\gamma_j - M_{ii}\sigma_i^{-1}\gamma_i = \sum_{j\neq i} M_{jj}\sigma_i^{-1}\gamma_j\\ &= \sum_{j\neq i} M_{jj}\prod_{k\neq i,j}\sigma_k.\end{aligned}$$

Putting this all together, we get

$$\bar{\mathbf{A}} = \mathrm{Sgn}(\det(\mathbf{U}))\mathrm{Sgn}(\det(\mathbf{V}))\mathbf{U}\mathbf{\Xi}\mathbf{V}^T,$$

with

$$\begin{aligned}\Xi_{ij} &= \begin{cases}\sum_{j\neq i} M_{jj}\rho_{ij}, & \text{if } i = j,\\ -M_{ij}\rho_{ij}, & \text{otherwise,}\end{cases}\\ \rho_{ij} &= \prod_{k\neq i,j}\sigma_k,\\ \mathbf{M} &= \mathbf{V}^T\bar{\mathbf{C}}^T\mathbf{U}.\end{aligned}$$

This allows us to compute second derivatives of the matrix determinant even for singular matrices. To handle degenerate matrices gracefully, we fuse everything from the computation of the log determinant to the final network output into a single TensorFlow operation, with a custom gradient and gradient-of-gradient that includes the expression above.

APPENDIX E: NONINTERACTING HYDROGEN CHAINS

At sufficently large separations, two systems become noninteracting. The energy of the combined system should be equal to the sum of the energies of the individual systems. We demonstrate this property for FermiNet on chains of well-separated hydrogen atoms of up to 10 atoms (Table VI).

TABLE VI. Chains of N hydrogen atoms at equal separations. The energy per atom is in excellent agreement with that of a single hydrogen atom.

	Energy / N (E_h)	
N	Separation: 10 a_0	Separation: 15 a_0
2	−0.5000023(9)	−0.50000021(5)
4	−0.4999977(6)	−0.4999991(2)
6	−0.499991(2)	−0.499990(1)
8	−0.499985(3)	−0.499993(2)
10	−0.499980(2)	−0.499989(1)

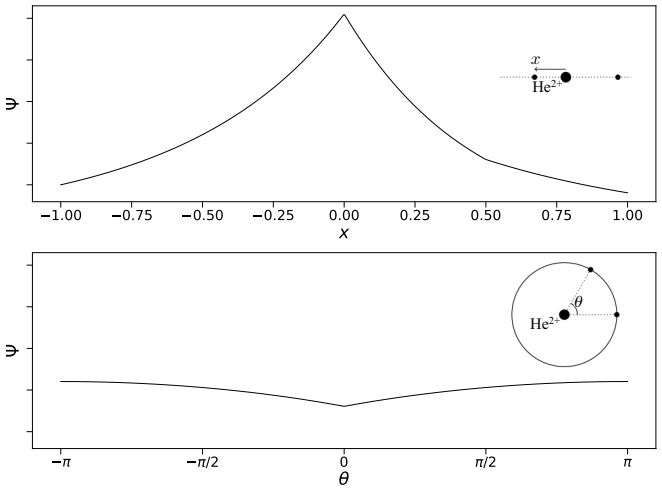

FIG. 11. Evaluation of the FermiNet wave function for the helium atom. The second electron is clamped at position $(0.5, 0, 0)a_0$ and the first electron is moved along the path $(x, 0, 0)a_0$, through both the nucleus and the second electron (top), and along the path $(0.5\cos\theta, 0.5\sin\theta, 0)a_0$, through the second electron (bottom).

APPENDIX F: ELECTRON-ELECTRON AND ELECTRON-NUCLEAR CUSPS

The derivatives of the wave function must be discontinuous when two electrons or an electron and nucleus coincide to cancel corresponding singularities in the Hamiltonian. Capturing these cusps correctly, especially the electron-nuclear cusp, is critical for accurately capturing correlation energy. Assuming the wave function is nonzero at these points, the cusp conditions specify the relationship between the wave function and its derivative to be

$$\lim_{r_{iI}\to 0}\left(\frac{\partial\Psi}{\partial r_{iI}}\right)_{\text{ave}} = -Z\Psi(r_{iI}=0),$$
$$\lim_{r_{ij}\to 0}\left(\frac{\partial\Psi}{\partial r_{ij}}\right)_{\text{ave}} = \frac{1}{2}\Psi(r_{ij}=0),$$

where r_{iI} (r_{ij}) is an electron-nuclear (electron-electron) distance, Z_I is the nuclear charge of the Ith nucleus and ave implies a spherical averaging over all directions.

Figure 11 shows FermiNet correctly describes the cusps for the helium atom. We estimate $\lim_{r\to 0}\frac{\partial\log|\Psi|}{\partial r}$ using Monte Carlo integration over spherical surfaces of radius $10^{-5}a_0$ centered on the helium nucleus and second electron, fixed at $0.5a_0$ from the nucleus, and obtain, where r_1 (r_{12}) is the distance between the first electron and the nucleus (second electron),

$$\left(\frac{\partial\log|\Psi|}{dr_1}\right)_{r_1=0,\text{ave}} = -1.9979(4),$$
$$\left(\frac{\partial\log|\Psi|}{dr_{12}}\right)_{r_{12}=0,\text{ave}} = 0.4934(1),$$

in excellent agreement with the theoretical values.

APPENDIX G: MOLECULAR STRUCTURES

Molecular structures were taken from the G3 database[71] where available. We reproduce the atomic positions for all molecules studied in Tables VII–XIII.

TABLE VII. Atomic positions for ammonia (NH_3).

Atom	Position (a_0)
N	(0.0, 0.0, 0.22013)
H1	(0.0, 1.77583, −0.51364)
H2	(1.53791, −0.88791, −0.51364)
H3	(−1.53791, −0.88791, −0.51364)

TABLE VIII. Atomic positions for methane (CH_4).

Atom	Position (a_0)
C	(0.0, 0.0, 0.0)
H1	(1.18886, 1.18886, 1.18886)
H2	(−1.18886, −1.18886, 1.18886)
H3	(1.18886, −1.18886, −1.18886)
H4	(−1.18886, 1.18886, −1.18886)

TABLE IX. Atomic positions for ethene (C_2H_4).

Atom	Position (a_0)
C1	(0.0, 0.0, 1.26135)
C2	(0.0, 0.0, −1.26135)
H1	(0.0, 1.74390, 2.33889)
H2	(0.0, −1.74390, 2.33889)
H3	(0.0, 1.74390, −2.33889)
H4	(0.0, −1.74390, −2.33889)

TABLE X. Atomic positions for methylamine (CH_3NH_2).

Atom	Position (a_0)
C	(0.0517, 0.7044, 0.0)
N	(0.0517, −0.7596, 0.0)
H1	(−0.9417, 1.1762, 0.0)
H2	(−0.4582, −1.0994, 0.8124)
H3	(−0.4582, −1.0994, −0.8124)
H4	(0.5928, 1.0567, 0.8807)
H5	(0.5928, 1.0567, −0.8807)

TABLE XI. Atomic positions for ozone (O_3).

Atom	Position (a_0)
O1	(0.0, 2.0859, −0.4319)
O2	(0.0, 0.0, 0.8638)
O3	(0.0, −2.0859, −0.4319)

TABLE XII. Atomic positions for ethanol (C_2H_5OH).

Atom	Position (a_0)
C1	(2.2075, −0.7566, 0.0)
C2	(0.0, 1.0572, 0.0)
O	(−2.2489, −0.4302, 0.0)
H1	(−3.6786, 0.7210, 0.0)
H2	(0.0804, 2.2819, 1.6761)
H3	(0.0804, 2.2819, −1.6761)
H4	(3.9985, 0.2736, 0.0)
H5	(2.1327, −1.9601, 1.6741)

TABLE XIII. Atomic positions for bicyclobutane (C_4H_6).

Atom	Position (a_0)
C1	(0.0, 2.13792, 0.58661)
C2	(0.0, −2.13792, 0.58661)
C3	(1.41342, 0.0, −0.58924)
C4	(−1.41342, 0.0, −0.58924)
H1	(0.0, 2.33765, 2.64110)
H2	(0.0, 3.92566, −0.43023)
H3	(0.0, −2.33765, 2.64110)
H4	(0.0, −3.92566, −0.43023)
H5	(2.67285, 0.0, −2.19514)
H6	(−2.67285, 0.0, −2.19514)

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Neural Information Processing Systems* (MIT Press, Cambridge, MA, 2012), Vol. 25, pp. 1097–1105.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Nature **529**, 484 (2016).

[3] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, in *Proceedings of the 34th International Conference on Machine Learning (ICML)* (JMLR.org, 2017), Vol. 70, pp. 1263–1272.

[4] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, Nat. Commun. **10**, 5024 (2019).

[5] K. Mills, M. Spanner, and I. Tamblyn, Phys. Rev. A **96**, 042113 (2017).

[6] A. V. Sinitskiy and V. S. Pande, arXiv:1908.00971 (2019).

[7] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller III, J. Chem. Phys. **150**, 131103 (2019).

[8] R. J. Bartlett and M. Musiał, Rev. Modern Phys. **79**, 291 (2007).

[9] M. Troyer and U. J. Wiese, Phys. Rev. Lett. **94**, 170201 (2005).

[10] M. Born and R. Oppenheimer, Ann. Phys. **389**, 457 (1927).

[11] G. H. Booth and A. Alavi, J. Chem. Phys. **132**, 174104 (2010).

[12] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Rev. Modern Phys. **73**, 33 (2001).

[13] R. P. Feynman and M. Cohen, Phys. Rev. **102**, 1189 (1956).

[14] M. Bajdich, L. Mitas, G. Drobný, L. K. Wagner, and K. E. Schmidt, Phys. Rev. Lett. **96**, 130201 (2006).

[15] R. Orús, Ann. Phys. **349**, 117 (2014).

[16] G. Carleo and M. Troyer, Science **355**, 602 (2017).

[17] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Phys. Rev. Lett. **121**, 167204 (2018).

[18] A. Nagy and V. Savona, Phys. Rev. Lett. **122**, 250501 (2019).

[19] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Phys. Rev. B **96**, 205152 (2017).

[20] L. Yang, Z. Leng, G. Yu, A. Patel, W.-J. Hu, and H. Pu, Phys. Rev. Research **2**, 012039 (2020).

[21] D. Luo and B. K. Clark, Phys. Rev. Lett. **122**, 226401 (2019).

[22] H. Saito, J. Phys. Soc. Jpn. **87**, 074002 (2018).

[23] J. Kessler, C. Calcavecchia, and T. D. Kühne, arXiv:1904.10251 (2019).

[24] J. Han, L. Zhang, and W. E, J. Comput. Phys. **399**, 108929 (2019).

[25] M. Taddei, M. Ruggeri, S. Moroni, and M. Holzmann, Phys. Rev. B **91**, 115106 (2015).

[26] M. Ruggeri, S. Moroni, and M. Holzmann, Phys. Rev. Lett. **120**, 205302 (2018).

[27] Since this manuscript appeared online, several other works using neural networks as Ansatz for continuous-space fermionic systems have appeared [72,73]. The first [72] also augments the typical Slater-Jastrow Ansatz with a deep neural network, while physical constraints like the cusp conditions are included explicitly. As the model has fewer parameters than ours, it is faster to optimize, but does not achieve the same accuracy. The second [73] represents a chemical system in second-quantized form using a given basis set, then fits a restricted Boltzmann machine to the ground state. This model is also able to exceed the performance of CCSD(T) *within* a basis set. However, it is less clear how easily this model extrapolates to the complete basis set limit. Our approach sidesteps the difficulty of choosing a basis set entirely.

[28] S. Zhang, *Ab initio* electronic structure calculations by auxiliary-field quantum Monte Carlo, in *Handbook of Materials Modeling: Methods: Theory and Modeling*, edited by W. Andreoni and S. Yip (Springer International Publishing, Berlin, 2018), pp. 1–27.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, in *Advances in Neural Information Processing Systems (NeurIPS)* (2017), pp. 5998–6008.

[30] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, Nature **577**, 706 (2020).

[31] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nat. Commun. **8**, 13890 (2017).

[32] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, J. Chem. Phys. **148**, 241722 (2018).

[33] J. Shawe-Taylor, Building symmetries into feedforward networks, *First IEE International Conference on Artificial Neural Networks, Conf. Publ. No. 313, London, UK* (IEEE, 1989), pp. 158–162.

[34] S. J. Chakravorty, S. R. Gwaltney, E. R. Davidson, F. A. Parpia, and C. F. Fischer, Phys. Rev. A **47**, 3649 (1993).

[35] B. K. Clark, M. A. Morales, J. McMinis, J. Kim, and G. E. Scuseria, J. Chem. Phys. **135**, 244105 (2011).

[36] E. Neuscamman, C. J. Umrigar, and Garnet Kin-Lic Chan, Phys. Rev. B **85**, 045103 (2012).

[37] R. Assaraf, S. Moroni, and C. Filippi, J. Chem. Theory Comput. **13**, 5273 (2017).

[38] L. Otis and E. Neuscamman, Phys. Chem. Chem. Phys. **21**, 14491 (2019).

[39] I. Sabzevari, A. Mahajan, and S. Sharma, J. Chem. Phys. **152**, 024111 (2020).

[40] J. Martens and R. Grosse, in *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (JMLR.org, 2015), Vol. 37, pp. 2408–2417.

[41] S. Amari, Neural Comput. **10**, 251 (1998).

[42] S. Sorella, Phys. Rev. Lett. **80**, 4558 (1998).

[43] J. Toulouse and C. Umrigar, J. Chem. Phys. **126**, 084102 (2007).

[44] P. Seth, P. L. Ríos, and R. J. Needs, J. Chem. Phys. **134**, 084105 (2011).

[45] W. Klopper, R. A. Bachorz, D. P. Tew, and C. Hättig, Phys. Rev. A **81**, 022503 (2010).

[46] W. Cencek and J. Rychlewski, Chem. Phys. Lett. **320**, 549 (2000).

[47] C. Filippi and C. Umrigar, J. Chem. Phys. **105**, 213 (1996).

[48] E. Giner, R. Assaraf, and J. Toulouse, Mol. Phys. **114**, 910 (2016).

[49] M. Dash, S. Moroni, A. Scemama, and C. Filippi, J. Chem. Theory Comput. **14**, 4176 (2018).

[50] T. Van Voorhis and M. Head-Gordon, J. Chem. Phys. **113**, 8873 (2000).

[51] H. Burton and A. Thom, J. Chem. Theory Comput. **12**, 167 (2016).

[52] D. Lyakh, M. Musiał, V. Lotrich, and R. Bartlett, Chem. Rev. **112**, 182 (2011).

[53] R. J. Le Roy, Y. Huang, and C. Jary, J. Chem. Phys. **125**, 164310 (2006).

[54] R. Gdanitz, Chem. Phys. Lett. **283**, 253 (1998).

[55] M. Motta, D. M. Ceperley, Garnet Kin-Lic Chan, J. A. Gomez, E. Gull, S. Guo, C. A. Jiménez-Hoyos, T. N. Lan, J. Li, F. Ma, A. J. Millis, N. V. Prokof'ev, U. Ray, G. E. Scuseria, S. Sorella, E. M. Stoudenmire, Q. Sun, I. S. Tupitsyn, S. R. White, D. Zgid, and S. Zhang (Simons Collaboration on the Many-Electron Problem), Phys. Rev. X **7**, 031059 (2017).

[56] A. Badinski, P. Haynes, J. Trail, and R. Needs, J. Phys. Condens. Matter **22**, 074202 (2010).

[57] M. Motta, C. Genovese, F. Ma, Z.-H. Cui, R. Sawaya, G. K. Chan, N. Chepiga, P. Helms, C. Jimenez-Hoyos, A. J. Millis *et al.*, arXiv:1911.01618 (2019).

[58] R. Q. Hood, M. Y. Chou, A. J. Williamson, G. Rajagopal, R. J. Needs, and W. M. C. Foulkes, Phys. Rev. Lett. **78**, 3350 (1997).

[59] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **8**, e1340 (2018).

[60] H. Flyvbjerg and H. G. Petersen, J. Chem. Phys. **91**, 461 (1989).

[61] C. Umrigar, M. Nightingale, and K. Runge, J. Chem. Phys. **99**, 2865 (1993).

[62] R. M. Lee, G. J. Conduit, N. Nemec, P. López Ríos, and N. D. Drummond, Phys. Rev. E **83**, 066706 (2011).

[63] R. Needs, M. Towler, N. Drummond, and P. L. Rios, CASINO: User's Guide Version 2.13 (2015).

[64] R. M. Parrish, L. A. Burns, D. G. A. Smith, A. C. Simmonett, A. E. DePrince, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, R. Di Remigio, R. M. Richard, J. F. Gonthier, A. M. James, H. R. McAlexander, A. Kumar, M. Saitow, X. Wang, B. P. Pritchard, P. Verma, H. F. Schaefer, K. Patkowski, R. A. King, E. F. Valeev, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, J. Chem. Theory Comput. **13**, 3185 (2017).

[65] A. DePrince and C. Sherrill, J. Chem. Theory Comput. **9**, 2687 (2013).

[66] D. Feller, J. Chem. Phys. **96**, 6104 (1992).

[67] T. Helgaker and W. Klopper, J. Chem. Phys. **106**, 9639 (1997).

[68] D. Petz and C. Sudár, J. Math. Phys. **37**, 2662 (1996).

[69] G. Mazzola, A. Zen, and S. Sorella, J. Chem. Phys. **137**, 134112 (2012).

[70] M. B. Giles, in *Advances in Automatic Differentiation*, edited by C. H. Bischof, H. M. Bücker, P. Hovland, U. Naumann, and J. Utke (Springer, Berlin, 2008), pp. 35–44.

[71] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. Pople, J. Chem. Phys. **109**, 7764 (1998).

[72] J. Hermann, Z. Schätzle, and F. Noé, arXiv:1909.08423 (2019).

[73] K. Choo, A. Mezzacapo, and G. Carleo, Nat. Commun. **11**, 2368 (2020).